March 1980 $2.00 U.S., $2.50 Canada

THE TRS-80* MAGAZINE

80 microcomputing™

THE magazine for TRS-80 users*

Printers Apprentice: Instant and accurate quotes. **Pg. 22.**

Biorhythms:
Your physical, mental and emotional state displayed. **Pg. 117.**

Investment Analysis:
Calculate your potential Return on Investment, in minutes. **Pg. 28.**

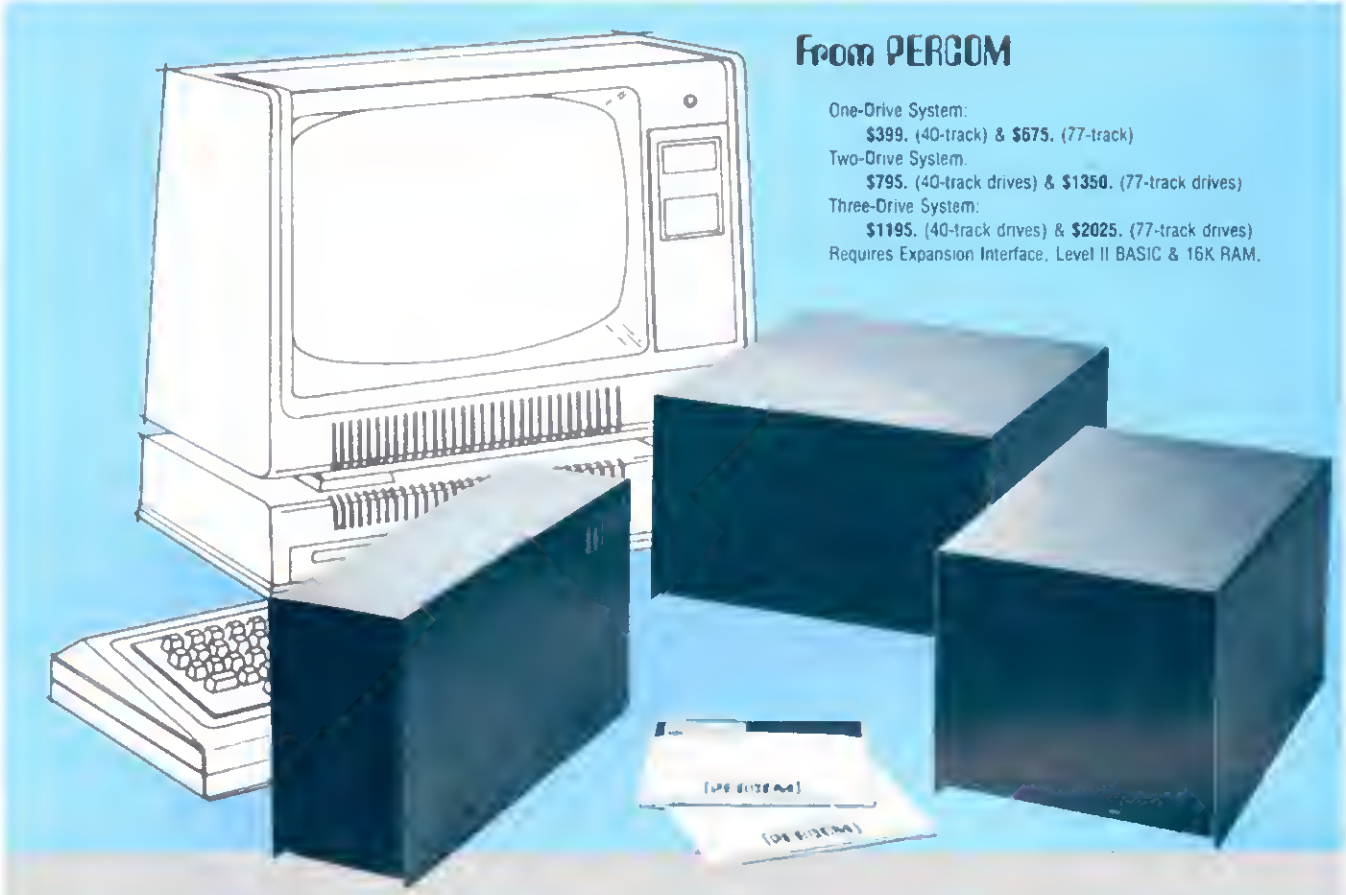
Artificial Intelligence:
A self modifying game program, that becomes unbeatable! **Pg. 55.**

KWIC Index:
Find information fast, with a technique developed by IBM. **Pg. 60.**

0 74470 65947 03

Plus:
14 Programs with complete listings.
Source Code for 4 Assembly Language programs.
More than 30 articles and columns.

*a trademark of Tandy Corporation



From PERCOM

One-Drive System:
$399. (40-track) & **$675.** (77-track)
Two-Drive System:
$795. (40-track drives) & **$1350.** (77-track drives)
Three-Drive System:
$1195. (40-track drives) & **$2025.** (77-track drives)
Requires Expansion Interface, Level II BASIC & 16K RAM.

Low Cost Add-On Storage for Your TRS-80*. In the Size You Want.

When you're ready for add-on disk storage, we're ready for you. Ready with six mini-disk storage systems — 102K bytes to 591K bytes of additional *on-line* storage for your TRS-80*.

- **Choose either 40-track TFD-100™ drives or 77-track TFD-200™ drives.**
- **One-, two- and three-drive systems immediately available.**
- **Systems include Percom PATCH PAK #1™, on disk, at no extra charge. PATCH PAK #1™ de-glitches and upgrades TRSDOS* for 40- and 77-track operation.**
- **TFD-100™ drives accommodate "flippy disks." Store 205K bytes per mini-disk.**
- **Low prices. A single-drive TFD-100™ costs just $399. Price includes PATCH PAK #1™ disk.**
- **Enclosures are finished in system-compatible "Tandy-silver" enamel.**

Whether you need a single, 40-track TFD-100™ add-on or a three-drive add-on with 77-track TFD-200™s, you get more data storage for less money from Percom.

Our TFD-100™ drive, for example, lets you store 102.4K bytes of data on one side of a disk — compared to 80K bytes on a TRS-80* mini-disk drive — and 102.4K bytes on the other side, too. Something you can't do with a TRS-80* drive. That's almost 205K bytes per mini-disk.

And the TFD-200™ drives provide 197K bytes of on-line storage per drive — 197K, 394K and 591K bytes for one-, two and three-drive systems.

PATCH PAK #1™, our upgrade program for your TRSDOS*, not only extends TRSDOS* to accommodate 40- and 77-track drives, it enhances TRSDOS* in other ways as well. PATCH PAK #1™ is supplied with each drive system at no additional charge.

The reason you get more for less from Percom is simple. Peripherals are not a sideline at Percom. Selling disk systems and other peripherals is our main business — the reason you get more engineering, more reliability and more back up support for less money.

In the Product Development Queue . . . a *printer interface* for using your TRS-80* with any serial printer, and . . . the *Electric Crayon*™ to map your computer memory onto your color TV screen — for games, animated shows, business displays, graphs, etc. Coming PDQ!

™ TFD-100, TFD-200, PATCH PAK and Electric Crayon are trademarks of PERCOM DATA COMPANY
*TRS-80 and TRSDOS are trademarks of Tandy [illegible]

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY • GARLAND, TX. • 75042

To order add-on mini-disk storage for your TRS-80*, or request additional literature, call Percom's toll-free number: 1-800-527-1592. For detailed Technical information call (214) 272-3421.

Orders may be paid by check or money order, or charged to Visa or Master Charge credit accounts. Texas residents must add 5% sales tax.

Percom *'peripherals for personal computing'*

TRS-80 TRS-80 TRS-80 TRS-80
TRS-80 TRS-80 TRS-80 TRS-80
TRS-80 TRS-80 TRS-80 TRS-80
TRS-80 TRS-80 TRS-80 TRS-80

The Complete Computer Company

HARDWARE
Centronics
TI 810
NEC 5530
Pertec
Shugart
MPI
CDC
Complete TRS-80 Mod I and Mod II Hardware
Percom

ENTERTAINMENT
Microsoft Adventure
Automated Simulations
Scott Adams
The Bottom Shelf

WORD PROCESSING
Magic Wand by Small Business Applications
Electric Pencil by Michael Shrayer
Special Delivery by Software Etc.
Merge Name & Address by SBSG

FROM THE LIBRARY
Learning Level II by David Lien
Basic Handbook
TRS Disk! and other Mysteries by H. C. Pennington
TRS 80 Software Source
Software Technical Manual by Houston Micro Computer Tech.
80-US
Kilobaud
80 Kilobaud

UTILITY & LANGUAGE
Small Systems
TRS 232
RSM Monitor
TRS Formatter
Racet
FMG
Lifeboat
Microsoft
Apparat

BUSINESS
Small Business Systems Group
Structured Systems Group
Lifeboat

*TRS-80 is a trademark of the Tandy Corporation.

To order by phone or for local dealer information call: **713/661-2005**
Texas residents add 6% sales tax • MasterCharge • Visa

HOUSTON MICRO-COMPUTER TECHNOLOGIES, INC. ⌐13

Home and Business Computer Specialists
5313 BISSONNET • BELLAIRE • TEXAS • 77401 • 713/661-2005

ALMOST PERFEC.

The MAGIC WAND™ is the most powerful, most flexible, most reliable, *most usable* word processing software available for a CP/M®-based computer.

That's not bragging. That's just telling it like it is.

The MAGIC WAND *is* the best word processing software ever written for a microcomputer. It can do more work in less time with higher quality than any other product you can buy.

The MAGIC WAND is a rock solid piece of software. The command structure is simple and logical and complete. We have not tossed in features without thought to the overall design of the package. Nor have we included any feature that is not thoroughly implemented. The programs are crash-proof and completely reliable.

And the system is supported by what we are told is the best user's manual ever produced for microcomputer software. It contains a step-by-step instructional program designed for the novice. The trainee uses sample files from the system disk and compares his work to simulated screens and printouts in the manual.

Support doesn't stop when you buy the package. As a registered user, you receive our bi-monthly newsletter which answers questions, reports upgrades and teaches new applications of the MAGIC WAND.

It's through a lot of hard work that we are able to offer you a product that is "almost perfect," but we aren't about to stop working until we can say that the MAGIC WAND *is* perfect.

Full screen text editing

The MAGIC WAND has probably the most responsive and easy-to-use editor available for either a serial or DMA terminal. It uses only single stroke control keys to give command and takes advantage of the special function keys on your terminal whenever possible. In addition, you can set up library files with coded sections that you can merge by section name.

Full text formatting commands

The MAGIC WAND allows you to set the left, right, top and bottom margins, page length, indentation, paragraph indentation, (incuding "hanging" paragraphs), text left flush, right flush, justified (two ways), literal or centered, variable line and pitch settings, variable spacing (including half lines), bold face, underlining (solid or broken), conditional hyphenation, sub- and superscripting. You may change any of these commands at run-time *without reformatting the file.*

Merging with external data files

You may access any external data file, with either fixed length or sequential records. The MAGIC WAND converts the record into variables that you define and can use like any other variable. Of course, you may use the data for automatic form letter generation. But you can also use it for report generation.

Variables

You may define up to 128 variables with names of up to seven characters. The current value of a variable may be up to 55 characters, and you may print it at any point in the text without affecting the current format. Although the MAGIC WAND stores the variables as strings, you may also treat them as integer numbers or format them with commas and a decimal point. You may increment or decrement numeric variables or use them in formatting commands.

Conditional commands

You may give any print command based on a run-time test of a pre-defined condition. The conditional test uses a straightforward IF statement, which allows you to test any logical condition of a variable. You may skip over unneeded portions of the file, select specific records to print, store more than one document in a single file, etc.

True proportional printing

The MAGIC WAND supports proportional print elements on NEC, Diablo and Qume printers. Other formatting commands, including justified columns, boldface, underline, etc., are fully functional while using proportional logic.

To order by phone or for local dealer information call: **713/661-2005**
Texas residents add 6% sales tax • MasterCharge • Visa

HOUSTON MICRO-COMPUTER TECHNOLOGIES, INC.

Home and Business Computer Specialists

5313 BISSONNET • BELLAIRE • TEXAS • 77401 • 713/661-2005 ↙13

small business applications, inc.

3220 Louisiana • Suite 205 • Houston, Texas 77006 • 713-528-5158

CP/M is a registered trademark of Digital Research Corp

80 microcomputing™ DATA

March 1980 Issue #3

Cover photography by Reese Fowler.

80 Microcomputing (ISSN # applied for) is published monthly by 1001001 Inc., Pine Street, Peterborough, NH 03458. Application to mail second class postage rate is pending at Peterborough, NH 03458 and at additional mailing offices. Phone. 603-924-3873/4. Subscription rates in the U.S. are $15 for one year and $40 for three years. In Canada $17 for one year and $46 for three years. In Europe please contact Monika Nedela, Markstr. 3, D-7778, Markdorf, W. Germany. In South Africa contact *80 Microcomputing*, P.O. Box 782815, Sandton, S. Africa 2146. Australian distribution by Katherine Thirkell, Sonotron Instruments, 17 Arawatta Street, Carnegie, Victoria 3163, Australia. All other foreign subscriptions $20 (one year only) surface mail. All U.S. subscription correspondence should be addressed to *80 Microcomputing*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send form #3579 to *80 Microcomputing*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737.

META TECHNOLOGIES

Fellow TRS-80 User:

I'd like to thank you for your interest in our products and tell you something about our company. Active in more than a dozen areas of the computer business, MTC's consulting and custom programming operations extend into five states. Although our work includes the more "traditional" minicomputers, the TRS-80 is no stranger. Members of our staff have worked on the TRS-80 since 1977. After five months of planning and preparation, MTC attacked the national market in the premier issue of 80 MICRDCOMPUTING (January, 1980).

Our marketing approach is simple. First, sell only top quality products. Some people question the quality of our Verbatim diskettes. We attract a lot of attention (and buyers) with our low prices, but we don't make our living selling diskettes. If you buy from us once, and have a chance to judge the quality and value of our products, we think you'll order again...and again. For example, MTC sells Apparat's popular NEWDDS†, complete with all utilities, at the same price as our competitors. But we include, at no additional cost, $39.90 worth of PROGRAMMING TOOLS! The cost to supply the tools is nominal...the value to you is significant. We make less profit on each sale, but we make more sales and more friends. Everyone makes out...except our competition.

Second, sell at below market price. When we offer a product at a lower price that meets or beats a competitor's product, we sell more than the competition. As our volumes go up, our cost goes down. Everyone makes out...except our competition.

Third, only offer useful products of lasting value. Every time you use one of our products, we want you to remember us fondly. Not being a fast-buck, basement operation, we want to have you as a customer for a long time. To ensure this, we intend to offer products that provide continual benefit...like diskettes, NEWDOS†, AIDS-II, etc. Everyone makes out...except (you guessed it!).

Fourth, make it as painless to order as possible. I don't know about you, but I hate ordering things through the mail. Give me a good, old TOLL FREE number anytime...which is exactly what we did. And three out of four of our customers use bank cards to charge their purchases. There is a single rate for shipping and handling (we normally ship UPS), whether your order is for one or a hundred-and-one items. Simple, convenient, no tricks, no hidden costs.

Finally, render fast and efficient service. Orders are tracked by computer, using our own AIDS-II system, and are usually filled within one business day. We pride ourselves on our rate of delivery.

In the coming months, we intend to use some of our ad space to feature letters from other members of our management staff (e.g. Customer Service, Product Engineering, etc.). We feel the better you get to know us, the more comfortable you'll feel dealing with MTC...and the less you'll deal with you-know-who.

MTC AIDS-II represents the essence of MTC's product development philosophy. Below are "testimonials" from four owners of the package. These are absolutely authentic statements and are typical of the comments we receive.

"The AIDS-II package is the most powerful data manager we've ever seen. The most complete and easiest to use!"
David Johnson, President, John-Tronics Security Co.
(Uses AIDS-II for client and prospect mailing lists, A/R, gun registration and alarm systems logs)

"It will do everything it's advertised to do plus whatever else your imagination allows. It is the best software value on the market today. AIDS-II is the most significant software development I've seen, and will have as large an impact on the marketplace as the Radio Shack TRS-80 did on the microcomputing market."
Vern Hall, V. Hall Insurance Agency (Nationwide Insurence)
(Uses AIDS-II for client and prospect mailing lists & follow-up systems, A/P, A/R))

"...the best system for the non-programmer I've ever used. It has an unlimited number of uses. I might have to buy another system just to have it on-line at all times... "
Robert I. Gross, CPA
(Uses AIDS-II for mailing labels, client reference system, for providing an audit trail to disburse funds to general ledger)

"...the most flexible and powerful system I've seen, especially with modules such as MAPS. The weakest part of the AIDS system is the Radio Shack Computer!"
L.G. Payne, Media Specialist, Strongsville High School
(Uses AIDS-II for mailing lists, tracking of audio-visual materials, experimenting with student attendance records)

We at MTC look forward to providing you with the quality products and service you deserve and should expect. If we offered you anything less, we wouldn't be Meta Technologies Corporation...we'd be you-know-who.

Sincerely,

Robert A. Fiorelli

Robert A. Fiorelli
President

TRS-80 and Radio Shack are registered trademarks of Tandy Corp.

MTC AIDS-II

Ailing information? Doctor it up with AIDS-II. This Automated Information Directory System is user-defined, features user-specified fields and print/display formats, conditional record selection, updating of fields within records, sorting by any combination of fields, and much more! Unique "windowing" capability allows directories of unlimited size. Window size is typically 200 or more records in 32K. Can be used for mailing lists, client reference reporting, appointment "calendars", inventory records and other information systems. Easy to use. Defining a system takes about a minute. Formatting a report or defining a custom label, less than 30 seconds. Sorting 200 records takes less than 5 seconds. Add "subsystems" for additional capabilities.

MTC AIDS-II $49.95
For Model II $79.95

CALCULATION SUBSYSTEM (CALCS)

Use for report generation involving basic manipulation of numeric data, such as quantity & cost computations, balances carried forward and columnar totals. Expands capabilities with respect to inventory, accounting and other numeric-based information systems.

MTC CALCS $24.95
For Model II $39.95

MAILING/INFORMATION LIST SUBSYSTEM (MAILS)

Use for report & label generation involving formatting of primarily non-numeric data, such as custom, "N-up" and "N-copy" label forms, index-type report formatting, and sub-fielding capabilities for selection & print. Expands AIDS-II with respect to client/product reference systems, mailing lists and other non-numeric information systems.

MTC MAILS $24.95
For Model II $39.95

RECORD/FILE ORGANIZATION EXECUTIVE (REFORGE)

Use for expanding, contracting or reformatting AIDS-II files. Convert random & sequential files to AIDS-II format. Use for converting mailing lists to AIDS-II. Includes MERGE & PURGE capabilities for combining smaller files into larger ones and removing duplicate records.

MTC REFORGE $24.95
For Model II $39.95

MTC TECH B.S.

Our exclusive Technical Bulletin Service reveals the inside story on the TRS-80® I & II. Sent by first class mail, bulletins are issued as the news breaks, not just once a month. Expensive, but worth it! No advertising or flyers, just pure Technical B. S.! Cancel any time - unused balance refunded. Free year-end subject index with 12-month subscription. Subjects have included "PEEK&POKE functions for Model II", "Machine Language Sort for String Arrays", "Tandy Marketing Plans", and "Level II Memory Locations". Subscriptions may be back-dated to obtain previous Bulletins.

1-Year Subscription
(Includes free index) $36.00
Monthly Subscription ..... $3.00/mo.

PEEK&POKE $14.95

Frustrated because PEEK and POKE have been removed from Model II BASIC?. Satisfy your curiosity with PEEK&POKE from MTC. Included are 8-bit and 16-bit (LSB, MSB) self-relocating machine language routines, instructions, and demo program.

MAKES EVERY BYTE COUNT

IN YOUR TRS-80® MODEL I OR MODEL II DISK SYSTEM

PROGRAMMING TOOLS

Buy 4, get 1 FREE
Any 5, $79.80
For Model II $119.80

★NEW★ ★NEW★
TDAM $19.95
For Model II $29.95

Having trouble with RANDOM FILES? With MTC's Table-Driven Access Method (TDAM) you'll never fret over FIELDing again. No knowledge of random access files is required. Insert the TDAM "interpreter" into any BASIC program and type in a few DATA statements describing the information in your files. TDAM does the rest! Reads and writes fields and records of any type (even compresses a DATE field into 3 bytes!). Features automatic file buffer allocation/deallocation, memory buffering, sub-record blocking/deblocking, and handles up to 255 fields per record. Super fast and super simple! Complete with TDAM interpreter, instructions and demo program.

DIVERGE $19.95
For Model II $29.95

Compares two BASIC program files, showing the differences between them. Identifies & lists lines which have been inserted, deleted, & replaced. Use for version control.

REBUILD $19.95
For Model II $29.95

Reorganize programs for adding program code, faster execution, readability. Much more than simple renumbering. Rearrange groups of statements within a program - automatically updates references to line numbers. Use with SUPERSEDE and MINGLE for maximum effect.

SIFTER $19.95
For Model II $29.95

Twelve in-memory high-speed sorts for use in any BASIC program: stable, non-stable, with/without tags, for numeric or string data. Random File Sort included. Some sorts written in machine code. Includes sort subroutines, demo programs and instructions. Relocate as needed with REBUILD.

SHRINK $19.95
For Model II $29.95

Makes Every Byte Count!. Make programs smaller and faster! Combines lines & removes unnecessary code including remarks, without altering program operation. Typically reduces program size 25% to 40%.

SUPERSEDE $19.95
For Model II $29.95

A "must have" for the professional programmer or the serious amateur. Probably one of the greatest time-savers available. Write programs in shorthand - change variable names - generate program documentation - use with REBUILD and MINGLE to build new programs from old ones.

MINGLE $19.95
For Model II $29.95

Merge two files (Program or Data) into a single file. Data files may be merged in ascending or descending sequence with the ordering based on a user-specified comparison field. A very handy utility for consolidating data files.

mTc-SHIRTS

HIGH-QUALITY, POLY-COTTON BLEND T-SHIRTS. White with Navy Blue neck and sleeve "ringers". MTC logo on sleeve. Top quality transfers of your choice on front.

Specify size (S,M,L,XL) and Transfer:

- META TECHNOLOGIES MAKES EVERY BYTE COUNT!
- DON'T TOUCH MY BITS!
- RAM it!
- MICRO LOVERS TAKE SMALLER BYTES!

mTc-SHIRTS $5.95

FILESORT $29.95
For Model II $49.95

Sort ASCII files by any combination of user-specified fields; ascending or descending, numeric or non-numeric keys. Optionally retains original file.

MAILSORT(Model I) $19.95

MAILSORT replaces existing sort in Radio Shack Disk Mailing List® system. Sort by any combination of fields, such as NAME, within CITY, within STATE. Hi-speed in-memory routine sorts 300 records in approximately 60 seconds. Minimum 32K recommended.

Single sided, Single density, Soft-sectored

DISKETTES

Verbatim 5¼-inch — MARCH SPECIAL*

$23.95 Box of 10

Hard-sectored (10-hole), Box of 10 ..... $26.95

8-inch FLOPPIES
Single-density, Box of 10 ..... $29.95
Double-density, Box of 10 ..... $39.95

PLASTIC LIBRARY CASES
5¼-inch or 8-inch diskette case $3.25
5¼-inch or 8-inch diskette case (when purchased with each Box of 10) ... $3.00

FACTORY FRESH, ABSOLUTELY FIRST QUALITY
Minimum order 1 box, NO order limit!
*Price in effect March 1 thru March 31, 1980

Complete for Model I with all utilities

NEWDOS +II

$99.95 by Apparat and MTC

40 TRACK VERSION $109.95

includes REF, RENUM, SUPERZAP, EDITOR/ASSEM., DISASSEM., DIRCHECK, and more! This is the original NEWDOS with all of Apparat's utility programs, plus any 2 MTC PROGRAMMING TOOLS (for Model I), of your choice.

Transfer PROGRAMS and DATA from MODEL I to MODEL II

TRAN-SEND

$49.95 by MTC

Requires MODEL II and MODEL I with disk & RS-232. Simple to use, not a kit - nothing else to buy. Complete with custom cable, 5¼" & 8" floppies, instructions. May be used over phone lines.

Custom Cable only $19.95

Suitable for use with Radio Shack® transfer program (ACT 0131)

All products guaranteed for replacement only. Prices, Specifications & Offerings subject to change without notice.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

QUANTITY DISCOUNT INQUIRIES INVITED

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $2.50 for shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

TO ORDER CALL TOLL FREE
1-800-321-3552
IN OHIO call (216)289-7500 (COLLECT)

✓20

META TECHNOLOGIES CORPORATION

26111 Brush Avenue, Euclid, Ohio 44132

800126
®TM TANDY CORP.

80 REMARKS

by Wayne Green

Sometimes I tend to assume too much and not communicate as well as I might. In the case of electronic mail I assumed that everyone else was as fed up with the increasing slowness and cost of the U.S. mail and that the value of sending mail electronically was self-evident. My error.

My concept of EM runs something like this. The time is well on its way when a microcomputer/terminal will be on most business desks and in most homes. I'll be able to type in a message, using the telephone number as an address. This message will then be sent to the addressee via telephone lines almost immediately.

My system will dial the number and if it's busy, it will cootinue to check the oumber every minute or so. When the line is free, the system will send a tone that prevents the phone from ringing on the other end and actuates the EM unit.

After the system receives a handshake signal, tell the other unit how many bits of information are forthcoming, send the message, await an okey, and bang both up. Time, at 1200 baud, perhaps one minute.

The EM unit on the other end will have a light indicating an awaiting message. This can be read when convenient and a response made . . . all within a minute or two, if needed.

Will Save Phone Calls

Such a system will not only speed up mail from several days or a week to a minute or two, but will also cut down on a lot of phone calls. I really hate to make phone calls. Often the other chap is on another line, or busy in a meeting, out to a late lunch, getting a haircut, or, perhaps, molesting his secretary.

The dollar loss in voice phone calls will be made up by an enormous increase in message billing, so the phone company will make out fine.

The U.S. Post Office will have to find something better to do, bless them.

While we are all waiting around for the government and Ma Bell to organize some sort of electronic mail system, it is my fiendish plan to get it going via microcomputers as quickly as possible. We'll have over a million micro systems out there by the end of 1980, and this will certainly be enough to get a service started.

Businessmen will be able to handle much of their own correspondence . . . at home, if they wish, just by re-routing the messages or having them repeated from the office. The present cost of correspoodence can be cut substantially where no typewritten copy is required, no secretary, no paper, no filing.

If a permanent copy is wanted, it can be filed on tape for later retrieval, put on microfiche, or even printed out and filed, if absolutely necessary. One of those relatively low cost 10,000 megabyte disks could replace file cabinets.

I know it would be heavenly sent, if I had such a system. If some irate subscriber had missed an issue, instead of calling me and raising Cain, he cnuld send a message. I would pass it along to the subscription department, a mile from my office. They would relay it to the fullfillment house in New York. The answer would come back directly to the subscriber, with an acknowledgement to our department and to me. All that would take just a few minutes instead of about three weeks.

I have asked several firms to design the hardware for just such an EM unit, and one has obliged. The interface will be strictly RS-232 and thus useful with almost anything, so we need software for all the popular micros, if any of you programmers are interested.

The program will have to word process, allowing the microcomputer to work as a correcting typewriter and to get it to tone or dial the needed number after a prompt. The system should be as transparent to the user as possible.

If you think you have the background for this, let me know. We are expecting prototypes of the interface unit soon and project a massive release of the system this coming summer. ■

80 APPLICATIONS

by Dennis Kitsz

A single screw on the case bottom is covered with a drop of tinted lacquer: it warns "Keep Out!" But you know a little electronics, and you've fixed your brother's TV . . . so into the TRS-80 you go, remembering the five cardinal rules of hobby electronics:

- Never get a *Technical Reference Handbook*. It's five more dollars they want after you've already put up hundreds.
- Pull everything apart to see what it looks like. Remove all socketed ICs and put them in a plastic bag.
- Test *everything*. There's nothing like a good torture test, so poke around with meter and scope probes and screwdrivers, and smack things with your knuckles. Be sure to do it all with the power on.
- Try experiments to see what happens. If the screen goes wacky, that's a sign you're doing something. Keep trying. Don't forget to twirl all the trimmer controls, too.
- If a part looks about the same, it probably fits; put it in. Corollary: Never read Thomas Hardy, Herman Melville, or specification sheets. They're all boring and playing "Space Potatoes" is more fun.

Three weeks later, you're still looking for a Radio Shack willing to take on the repair work. Two months after that, you've got it back, plus a bill for over a hundred dollars.

Poking, Prodding, Thumping

Of course, I am suggesting that the TRS-80 is a pretty sophisticated electronic toy and not very responsive to poking, prodding or thumping. It can succumb with devilish quickness and quietness.

If you've decided that your TRS-80 needs a change or addition, and you intend to do it, your chance of success can be high. But do buy and read the *Technical Reference Handbook*. You may not understand it all, but even a general feeling of how, why and when the machine performs its activities is worth the time spent.

If you're installing someone else's hardware addition or modification, read the documentation carefully. Understand what the change does and how it works.

Always open the case with great care. The boards are made up of hundreds of delicate circuit traces that can be broken with a scratch, and there are several rather unpleasant interconnection cables that can unexpectedly snap loose. Leave sensitive ICs and cables in place.

I took my TRS-80 apart the very day I bought it . . . and broke the Level II interconnect cable. Two weeks, ten dollars.

Leave the power off until testing time, and then reseal the case. It's no fun to discover that the IC you piggybacked or the DIP switch you put in place of a jumper works just beautifully, but putting on the cover crushes it into the circuit board, sending the computer into a microfrenzy.

Never just "try it." There may be more than one way of achieving a given change, but avoid

Continued to page 10

BASIC SOFTWARE LIBRARY

NOW ★ 10 ★ Volumes and Growing

IS SPONSORING A

$10000.00 Give Away

And unlike others we are giving a large portion in CASH that you Don't have to spend with us. You are eligible NOW! You may already be entered. More details in future advertisements.

We have over 100,000 in circulation since 1975 and we are still around and That's more than Anyone else can say. We used to sell hundreds of programs individually, the programs in Volume X were sold for several years at over $10,000, in Volume III for over $6,000 but a few years ago we decided to promote software to the mass public and it was an instant success. We are still several years ahead of our time in our marketing concepts as well as our products and we are going to be making another major change in the market. We are going to offer our programs in cassette form. NOT just one of two programs like everybody else. But a book full of programs for just $9.95.

For Homeowners, Businessmen, Engineers, Hobbyists, Doctors, Lawyers, Men and Women

Vol. I $24.95

Business & Personal Bookkeeping Programs

Bond
Building
Compound
Cyclic
Decision 1
Decision 2
Depreciation
Efficient
Flow
Installment
Interest
Investments
Mortgage
Optimize
Order
Pert Tree
Rate
Return 1
Return 2
Schedule 1

Games & Pictures

Animals Four
Astronaut
Bagel
Bio Cycle
Cannons
Checkers
Craps
Dogfight
Golf
Judy
Line Up
Pony
Roulette
Sky Diver
Tank
Teach Me

A. Newman
J.F.K.
Linus
Ms. Santa
Nixon
Noel Noel
Nude
Peace
Policeman
Santa's Sleigh
Snoopy
Virgin

Vol. II $24.95

Binomial
Chi-Sq.
Coeff.
Confidence 1
Confidence 2
Correlations
Curve
Differences
Dual Plot
Exp-Distri
Least Squares
Paired
Plot
Plotpts
Polynomial Fit
Regression
Stat 1
Stat 2
T-Distribution
Unpaired
Variance 1
Variance 2
XY.

APPENDIX A

Beam
Conv.
Filter
Fit
Integration 1
Integration 2
Intensity
Lole
Macro
Max. Min.
Navaid
Optical
Planet
PSD
Rand 1
Rand 2
Solve
Sphere Trian
Stars
Track
Triangle
Variable
Vector

Vol. III $39.95

Billing
Inventory
Payroll
Risk
Schedule 2
Shipping
Stocks
Switch

Vol. VI $49.95

Mini-Ledger
Payroll
A/R
Inventory
Peprec.
Ledger

Vol. VII $39.95

Chess
Medbl
Wdproc
Utility

Vol. IV $9.95

Bingo
Bonds
Bull
Enterprise
Football
Funds 1
Funds 2
Go-Moku
Jack
Life
Loans
Mazes
Poker
Popul
Profits
Qubic
Rates
Retire
Savings
SBA
Tic-Tac-Toe

Vol. V $9.95

Andy Cap
Baseball
Compare
Confid 10
Descrip
Differ
Engine
Fourier
Horse
Integers
Logic
Playboy
Primes
Probal
Quadrac
Red Baron
Regression 2
Road Runner
Roulette
Santa
Stat 10
Stat 11
Steel
Top
Vary
Xmas

APPENDIX B

Vol. VIII $19.95

1040-Tax

Balance

Checkbook

Instal 78

Oeprec 2

APPENDIX C

Favorites

Vol. IX $19.95*

Auto Cyphar
Hurrirec IO
Lorane Map
Navigate
Omega
Patterns
Radar
RDF

Vol. X $89.95*

Intro.
A/R
A/P
Mer Inv
Check
Assets
Payroll
Bal Sh
P/L
Year End
Data Base
Tax Up
Basic St.

*10% Discount — Offer good til April 30, 1980

Volume III—Part 1—Billing Inv. Pyrl.
Volume VII—Part 1—Chess
—available on TRS-80 level II compatible cassette $9.95
Add $1.00 per tape Post./Handling

Volume VI — Disk programs are compatible with TRS-80 disk basic
The disk programs in Volumes VI, VII and X are written in (CP/M) M Basic and Disk Extended Microsoft Basic. Other programs written in 8K Basic.

Add $1.50 per volume handling, all domestic shipments sent U.P.S. except APO and P.O. Box which go parcel post. Foreign orders add $6.00/volume for air shipment and make payable in U.S. dollars only.

AVAILABLE AT MOST COMPUTER STORES
Master Charge and Bank Americard accepted.

Unlike others we have NOT raised our prices in five years

KEMCO, LTD. ↙157
P.O. Drawer 2208G Petersburg, VA 23803
24 hour Hot Line (800) 528-6050
ext. 1184. In Arizona (800) 352-0452

OVER 116,000 IN USE TODAY

80 APPLICATIONS

Continued from page 8

experimenting with a disemboweled TRS-80 before you. If what you have installed doesn't work, don't jumper connections or rotate controls. You did something wrong, so admit it, and recheck or rethink everything. Test carefully, using good equipment and proper test clips.

Read Those Spec Sheets

Last of all, when doing your own design or substituting parts, read specification sheets. This warning (as well as this entire column, I must admit) derives from my own experience. The TRS-80's designers were kind enough to leave some blank area that could contain an extension of the Level II ROM. Delighted at the prospect, I set about designing a small ROM board to include utility programs of interest.

I decoded the address of the area, and hand-wired a small board that contained various ICs and $42 worth of 1702A EPROMs. These handy little circuits were relatively inexpensive, and they could be individually erased for future program changes.

It didn't work. I had considered everything but *speed*. All the mysterious talk about access time and X number of nanoseconds came into focus.

A quick look at the TRS-80 clock revealed a speed of 1.66 MHz, which meant one cycle every—quick calculation—600 nanoseconds. Turning to the spec sheets, I found the 1702A had an access time of—disappointment edging toward anger—900 nanoseconds.

Even veteran experimenters need to be reminded that digital circuits have a low tolerance for abuse or marginal design. On the other hand, well-designed hardware additions or modifications are very much a part of the community of TRS-80 hobbyists, and that's one way of making these microcomputers respond to human needs.

Additional Notes

Additional notes on *Faster! Faster!* (Feb. *80 Microcomputing*) for those of you with disk drives. Howard Batton of Auburn, WA, bravely went ahead with the modification. He says:

"I went the whole route the first time . . . as you suggested, the disk won't power up in the high speed mode, so I had to reverse the Q and Q' leads. As it stands now, after only a couple days to try the system, I can use the high speed most of the time, including disk reads and writes from BASIC. The DOS commands, however, don't want to work properly, at least not all of the time. Some of them, e.g. format, copy boot, don't work at all in high speed mode. So far, I haven't lost any programs or wiped out any diskettes, so I count the high speed mod a success."

So you disk owners will want to power-up at the lower speed. As noted, reverse the leads from Z7 to ZSPEED. Now OUT 254,0 gives you low speed, OUT 254,1 is high speed. Very many thanks to Howard. ■

CLUB 80

by Ross Wirth

I am happy to hear from you to learn your ideas and thoughts on the content of this column. Please send your comments to me at 15906 E. 96 St. N., Owasso, OK 74055. An SASE would be appreciated for personal replies.

Newsletter Review: Chicatrug News

Chicatrug is a TRS-80 User Group that meets every month on the third Wednesday from 6:00 to 9:00 PM at 203 N. Wabash, Room 2102 in downtown Chicago. Their monthly newsletter for January was ten pages long. The breakdown of the newsletter was 1½ pages of short notes and new product reviews, 1 page of meeting announcements, 4 pages of ads, and 3½ pages of articles written by club members. These articles include a review of programs that have sound output, a machine language program for drawing a pinwheel and a line between two (x,y) points and a good explanation of how to use VARPTR in your programming.

Chicatrug News is published monthly by Emmanuel B. Garcia, Jr. & Associates, 203 N. Wabash, Room 2102, Chicago, IL 60601. Annual subscription rate is $12.

Programming Hints

One topic that everyone seems interested in is short programming tricks that they can incorporate in their programs. For those who are interested in such goodies I will present one or two each month. I'll try to give credit to the people who bring them to my attention. The original creator will also be credited, if known.

Loss of information on the screen when entering data: While playing a game you are prompted to enter some information. You type the information in and hit ENTER and before your eyes a line of the screen is blanked out as the cursor moves to the next line. The problem: you cannot control the cursor moving to the next line.

Solution: Use INKEY$ for entering information to your program. This will prevent the loss of information on the screen and will keep the cursor at its present location.
Try this example.

```
100 CLS
110 FOR I=0 TO 47:SET(56,I):NEXT I
120 PRINT @ 339,"NUMBER?";
130 X$=INKEY$:IF X$="" THEN 130
140 X=VAL(X$)
150 X2=X*X
160 PRINT @ 350,X;" SQUARED IS ";X2;
170 GOTO 120
```

Notice that VAL was used to change the character input to a numeric value. If a non-numeric character is entered VAL will return a zero (0). Knowing this you can check for non-numeric input in the data validation section of your program.

Note: this works for single digit input. A later hint will deal with multiple digit input using INKEY$.

Arrows as input to a program: (brought to my attention by Greg Perry, Tulsa, OK). Every key on the keyboard is available for entering data to the computer, including the arrow keys. Instead of using U, D, L, R for up, down, left and right, the arrows can be used directly. Useful applications include games and word processing. Try the following example to see how this works.

```
100 CLS
110 IF PEEK(14400)=8 THEN PRINT "UP"
120 IF PEEK(14400)=16THEN PRINT "DOWN"
130 IF PEEK(14400)=32 THEN PRINT "LEFT"
140 IF PEEK(14400)=64 THEN PRINT "RIGHT"
150 GOTO 110
```

Try this use of arrows in the next program that needs directions as input.

Short Notes

Recordings of programs on tape will last longer if you press the STOP button after loading the program into your computer. Leaving the PLAY button depressed keeps constant pressure on the tape, the last part of which contains the last part of your program. This continued and repeated tension on the tape can eventually cause drop out of data. (And who wants that to happen!)

• • • • • • • •

Last month I mentioned the formation of groups of individuals for the purpose of information exchange and education. My role in this activity is to match individuals with similar interests. Some additional areas of interest are health care, education, war-gaming, word processing and scientific applications.

Each of these groups is in need of a coordinator. If you wish to join one of these groups or wish to serve as a coordinator please drop me a note and I'll make sure you get together.

• • • • • • • •

Computer hardware costs are still dropping and general software is becoming more powerful and available. In the past software was given away to sell computer hardware. In the future will the hardware be given free with the purchase of a software package?

As hardware becomes, cheaper, it is financially more feasible to build bigger, faster and more complex systems then ever before. Send me your thoughts on the future of hardware. ■

Software for the TRS-80

80 INPUT

Spinner Rebuts

The first issue of *80 Microcomputing* came Monday, and it was hard to wait until the dinner dishes were done and I was free to retreat to my favorite chair to spend several hours reading about my favorite subject: the TRS-80. In general I was very pleased with the initial issue of *80 Microcomputing*. I was particularly impressed with the broad range of TRS-80 operation/utilization issues the magazine articles covered.

As a new disk spinner I was naturally attracted to the article by William O'Brien "A Disk Primer." While I thought it was well written, I did have two objections/observations about points mentioned in that article.

My first objection is to the comment (page 130, column 2, the second paragraph) which states " . . . Level II tapes are not compatible with the disk operating system." That is just not true.

Mr. O'Brien has probably confused the Level I-Level II tape incompatibility problem with the problem of real time clock interference under Disk BASIC. A real time clock is nice, but it does wreak havoc with tape input and output operations. In fact, as long as the clock is operating normally you get garbage about 80 percent of the time during tape operations. The solution is to turn the clock off before tape operations. That is accomplished by issuing the command CMD"T" just before tape operations. After operations are complete, the command CMD"R" will restart the clock's operation. The process only seems complex the first couple of times you do it, after that it becomes second nature.

(Some disk users have had interference problems from the clock with their disk input-output operations, but I understand that Percom has a simple, cheap plug-in board that solves that particular problem.)

The key point is that those Level II tapes will still work after you add a disk to your system.

The second observation I have is regarding a comment Mr. O'Brien makes further on:

> "Since you TRS-80 users must have the Radio Shack DOS disk present in drive 0, there is not much room left for storage (for instance, RS's 2.2 DOS leaves only 18 kilobytes of storage of a possible 89.6 kilobytes). That means two disk drives."

It does not mean two disk drives. It means you have to be a bit creative. The solution lies in constructing a minimal operating system. A minimal operating system is one that contains only those DOS files you absolutely need for normal operations. The other files are killed, freeing the space they occupied for other use. Using TRSDOS 2.3 as my example, it works something like this. As your DOS disk comes from Fort Worth it has 38 free files and 21 grans of disk space you can use. This translates into about 26 kilobytes of space. If we kill the following files, it is possible to more than double the free space:

```
FORMAT/CMD. FORMAT  (THE ".FORMAT" IS
                     THE PASSWORD)
BASICR/CMD. BASIC   (THE ".BASIC" IS
                     THE PASSWORD)
BACKUP/CMD. BACKUP  (THE ".BACKUP" IS
                     THE PASSWORD)
TEST2/BAS
TEST1/CMD
TAPEDISK/CMD
GETDISK/BAS
GETTAPE/BAS
DISKDUMP/BAS
```

After you have killed these files, your drive 0 diskette will have 47 free files and 52 free grans of space (roughly 66 kilobytes of free space). You can do a great deal with 66 kilobytes of mass storage! The most profitable files to kill—in terms of the space they free are:

```
TEST2/BAS     (FREES 15 KILOBYTES)
BASICR/CMD    (FREES 6 KILOBYTES)
FORMAT/CMD    (FREES 3.8 KILOBYTES)
BACKUP/CMD    (FREES 3.8 KILOBYTES)
```

Interestingly enough, these are also the files that I find I don't need to have on every diskette I own. But whatever mix of DOS files you have on your drive 0 diskette, the key point is that there is a great deal more mass storage capacity on that diskette than a cursory glance would indicate. I'm no great shakes as a programmer, but I've done a lot of work with the storage space on my drive 0 diskette.

I'm looking forward to the next issue of *80 Microcomputing*.

James M. Kenderdine
13420 East Cedar Lane Road
Norman, OK

System Crashes

I just received the first copy of *80 Microcomputing* in the mail this week and I like what I see! I think I held my breath long enough waiting for someone to come out with a magazine devoted to just the TRS-80.

You mentioned in the magazine about random crashes when the interface is used. I have a 32K system: one disk system is hooked up and another has been ordered. I also have a PR-40 printer. As for system crashes, yes I have had some with the interface turned on. I sent the interface in to the Radio Shack service center and they sent it back saying "no problem!"

Well, it crashed again, this time with the disk running! Lucky all the files were CLOSED or it would have been a mess! I had to push the reset and boot DOS back into the machine.

As for software I write much of my own and copy some out of books and debug them so they will work in my machine. Some programs "written for the TRS-80" have to be debugged in order to work in my machine!

One program in particular left a space in the file spec and you don't leave blank spaces in your filespecs because the information after the blank space is ignored. I had to change from M$ + M1 to M$ + A$, then to get the numeric value of A$ I used VAL(A$) to get the numbet back for my IF branches. This was the only way I could get rid of that blank space!

I am planning to add another printer to my system, a full size printer but I am going to keep the PR-40 on line for use when a 40 column printer is more practical. I am going to have it switched from my control console.

James Weisjahn
Box 396
Medford, MN

Nobody's Perfect

I enjoyed the first issue of your new publication and urge you to keep up the good work. As a TRS-80 Level II owner the articles were very useful, however, I would like to bring the following points to your attention!

"Basic BASIC Renumbering," p. 82.

If line 10070 is changed to: 10070 IF<>255 THEN L = L − 256: H = H + 1 the program will renumber all of the lines in multiples of 10 instead of the "6" spacing every 25th line.

It is better to add H = 0 to line 10000 for safety in case H has been pre-initialized. Incidentally, running this program renumbers its own first three lines!

"NEW Restored," p. 84.

I had problems with the FIXNEW program and finally figured out that it was O.K. if after running it, it was immediately followed by a RUN.

If you started with LIST to see if it worked, it went to hell! My TRS-80 is new (August), maybe it's different. I discovered it needed locations 40FD/FC and 40FD/FE loaded with the contents of register HL. Doing this necessitates moving the starting address back a few bytes otherwise you'll go past 4FFF. With my change

the program ran fine.

"Get T-BUG High," p. 118.

This is a great program for idiots like myself who are not too well up on machine language—once it is de-BUGged. I found that Figs. 2 & 3 had their titles switched and that the BASIC driver had an error on line 30. Line 30 should switch the program to line 70, not 75!: then it all works fine.

Terrific! I used this shift (now called TBUGHI) to figure out the FIXNEW program. Don't be discouraged! I'm looking forward to Issue 2.

Keith Walker
1075 Brush Hill Lane
Lake Zurich, IL

I Was Cheated

Radio Shack's Mierochess cheats to win.

There are certain conditions that can be reached during normal game play with MICROCHESS 1.5 that will allow the computer to move illegally—cheat. These conditions are:

1. The computer plays the black pieces.
2. The level of play is set for IQ = 3.
3. The computer's king must be in check and it cannot simply be moved out of check.
4. One of your pawns must be blocking movement of one of the computer's unmoved pawns.
5. Your piece forcing check must have its line of action cross the space immediately behind your pawn that satisfies condition 4.
6. The nature of the game will be changed if the computer cheats.

I stumbled on this error while playing against the machine and have since verified each of the conditions listed. Condition 6 is interesting since apparently, if cheating doesn't gain an advantage, the machine won't cheat, even if the rest of the conditions are met.

The sequence for the game I was playing when the error was discovered is listed below.

MOVE	WHITE	BLACK'S RESPONSE
1	D2-D4	G8-F6
2	B1-C3	F6-G4
3	E2-E4	G4-F6
4	E4-E5	F6-G8
5	F1-C4	O7-D5
6	C4-D5	C7-C6
7	O5-C4	D8-A5
8	O1-F3	B6-D7
9	F3-F7	E8-D8
10	E5-E6	D7-B6
11	F7-F8	D8-C7
12	C1-F4	E7-E5

Note that black's 12th move is illegal in that the computer has moved a blocked pawn.

I would appreciate hearing from anyone interested in this problem or its solution.

Mike Tollerton
RD#2
Blossvale, NY

More on Computer Music

It is about time somebody cared enough for the TRS-80 to create a "real" magazine for it! I have seen several different ones—from the newspaper type (which are big jokes), to your sister magazine *Kilobaud*. This is by far the best and hopefully most successful endeavor for the computer hobbyist.

I was glad when I read Dennis Kitsz's *80 Applications* column and saw that he too has a musical synthesizer! I would certainly hope to see in these pages a few articles on computer music, and especially applied to our TRS-80. I know for a fact that my Steiner-Parker "Synthacon" has been linked to a National Semiconductor PACE system. If it can be done with a slightly esoteric system, then why in the hell can't it be done with my TRS-80?

On the subject of computer music, when is somebody going to come up with one for the TRS-80 that is humanly engineered? I have both Shack's "Micro Music" and Mad Hatter's "Musicmaker" and both are painfully laborious and mundane. How about one that lets you enter notes as shown on a musical staff and displays each note on the staff as they are being played? Are we going to let Apple get the best of us? Or how does one get to know Max Mathews then?

I would certainly hope to see a few articles on computer music.

Now, how about some simple articles or teaching series on Assembler language. Even though a friend of mine (with a TRS-80) is attempting to help me, I can't seem to get past the NOPs and POKEs. HELP!!!! I'm sure I'm not alone in the boat.

Now that you know where I'm at, I'll be anxiously awaiting my wonderfully speedy (choke! choke!) mail service for what *80 Microcomputing* is going to do to entice me NOT to give up after my first year!

Mark S. Lucas
724 E. Mulberry St.
Lancaster, OH

Program Size?

I just received Issue #1 and am impressed at the quality of both the magazine and its content.

After reading Rod Hallen's "Software Review," I was left with the nagging question which never seems to be answered until you purchase the software: How big is the program?

Especially with a program like Line Renumbering or GSF which is intended to run with BASIC programs also resident in memory, it is important to know how big the utility program is. Because if a Line Renumbering program is too large it *cannot* be used to renumber a large BASIC program.

I'd suggest that you add program size as a part of all reviews.

Dan Walther
2465 Tyler Road
Birmingham, AL

Dogeared Record

80 is superb. Your first issue must be recorded as the greatest single issue of a computing magazine ever.

I skim a magazine, dogearing an ad or article that I want to return to. Issue #1 hit a recoed 17 dogears.

Your advertisers will be pleased to know that they have collectively plucked eight orders from me.

As Durocher said, "Push your luck."

Sean Tomlin
2108 Hurly St.
Glenview, IL

80 DEBUG

Getting T-BUG HI

Received a call today from a reader in reference to a problem he was having with the article I submitted. There is a typo error indeed, which I did not catch in the manuscript.

Line 30 of the BASIC program should end THEN 70 rather than 75. The peek address does not get incremented as the program currently reads. It was my error in submission and proofing.

Irwin Rappaport
24 Hemlock Hill Rd.
Upper Saddle River, NJ 07458

Errata

It has been brought to my attention that I made a major goof in the printout of the SORT program that I submitted for publication. "Sort 80K in 6K!" Jan. *80 Microcomputing*. Making the following changes to the program as listed will provide proper operation.

```
250 LSET D1$ = D3$:PUT 1, P8:GET 1,P3:O3$
 = D1$
:P8 = P3
270 LSET D1$ = D4$:PUT 1, P9:GET 1,P4:D4$
 = O1$
:P9 = P4
320 MID$(O3$,1 + N6,N1) = T2$
330 MID$(D4$,1 + N7,N1) = T1$
370 IF P3 <> P4 THEN 440
```

Sorry to have inconvenienced you and your readers.

D. E. Fitchhorn
3504 Piermont Dr. N.E.
Albuquerque, NM

Continue to page 16

80 ACCOUNTANT

by Michael Tannenbaum C.P.A.

Listing an inventory is the most common application for small computers such as the '80. Inventories are created for all sorts of items, from credit cards to phone numbers, and shuffled in many different ways. People in business are always looking for new ways to view their inventory, by type, by size, by color, by price . . . the list is endless.

In addition to a file maintenance program for adding, deleting or modifying data, the inventory software system usually contains a report generating program and a sorting program. These programs are used in combination to resequence the file according to key words, while generating hard copy or visual reports on the monitor.

While the forms of various inventory systems may be similar, the reports generated are not. Inventory can be controlled by type, value, age, demand, quantity, location or combinations of the above. The point is that you must define your needs before purchasing a system.

The Valuation

As an accountant I examine an inventory's valuation. Methods of valuation include:

- FIFO—First In First Out
- LIFO—Last In First Out
- Average Cost
- Specific Identification
- Retail Method

Regardless of which method you choose, both beginning and ending inventories must be valued under the same one. Different valuation methods yield different profit pictures. Let's consider Tables 1 & 2.

The LIFO method results in the lowest gross profit of all the methods illustrated. Under this method, the income statement includes the most recent costs (Last In First Out) and the balance sheet is left with the earliest costs. As a result, during inflationary times, this method results in a lower income and thus a lower income tax.

But, as you no doubt anticipated, there is no such thing as a free lunch. Should the closing inventory units drop lower than the opening inventory units, the low cost units are included in the income statement and the tax deferral is reversed with a vengeance.

Radio Shack has issued several Inventory maintenance programs. One of their earliest efforts was ICS (Inventory Control System Catalog #26-1553). This was followed by the manufacturing inventory control system released at the end of 1979. Although the two systems are significantly different, they both represent usable packages which will be supported by the vendor.

The ICS system was intended as an inventory tool for a merchant who purchases inventory in a saleable condition and marks it up for resale —just like a Radio Shack store. Because the designers of the system were familiar with their own operation, they obviously used it as a model when setting up ICS parameters. This observation should not be construed as a deficiency, however, it is important to understand the designer's intentions when considering the system for your use.

The ICS parameters include the following:

- Utilization of a stock number referencing scheme that requires merchandise to be coded with a separate prefix and suffix. The purpose of this scheme is to group related items yet permit item identification within the group. Explanation of the purpose of this procedure is not included in the documentation which accompanies the system.
- Preparation of inventory reports in which cost figures can be suppressed.
- Display of a cost/retail relationship on the inventory report. Such information is valuable only if the firm has a fixed selling price or the inventory controlled is offered for sale to outsiders.
- Batching sales prior to posting in a holding file until an update run is made. As a result, "available to sell" or "use" status is not available immediately.

A prominent feature of the system is the use of a "reorder point" that can identify items in need of replenishment. Defining your optimal reorder points is a major financial goal and one that can justify the purchase price of a computer, related peripherals and software. But establishing your reorder points requires careful consideration.

Key elements in the calculation of a reorder point usually include the following:

- Rate of usage or sale;
- Delivery time required by the vendor and associated transport after a purchase order is placed;
- Minimum lot size required by the vendor when placing an order;
- Definition and quantity of your firm's "Stock Out" philosophy by merchandise type.

In addition, consider your capacity and any

	UNITS	PER UNIT	EXTENDED DOLLARS
Sales	250	$5.00	$1,250.00
Invent.—Beg.	100	2.00	200.00
Purchases (1)	50	2.00	100.00
(2)	50	2.50	125.00
(3)	50	3.00	150.00
(4)	50	3.50	175.00
(5)	50	4.00	200.00
Invent.—End	100	?	

	FIFO	%	LIFO	%	AVG. COST	%	SPECIFIC I.D.	%
Sales	$1250	100	$1250	100	$1250	100	$1250	100
Cost of Sales								
Invent. Beg.	200		200		200		200	
Purchases	750		750		750		750	
Avg. Cost	950		950		950		50	
Less Invent. End*	375		200		272		350	
Cost of Sales	575	46	750	60	678	54	600	48
Gross Profit	$ 675	54	$ 500	40	$ 572	46	$ 650	52
*Invent. Comp.	50 @ 4.00 50 @ 3.50		100 @ 2.00		100 @ 2.72		50 @ 4.00 15 @ 3.00 5 @ 2.00 10 @ 2.50 20 @ 3.50	

Tables 1 & 2.

financing requirements. Above all consider your rate of consumption. ICS does not maintain this statistic. As a result, you will have to establish a separate sales accumulation system.

Adjusting the Record

Another puzzling problem relates to the accumulation of inventory cost, selling price and on-order totals. As data is entered, the product of the units and the price field is used to adjust a total record which is carried forward each time a posting session is completed. These totals are displayed when system status is requested or at the bottom of reports that display the inventory in its entirety. However, when each item is multiplied times its unit cost and retail, the sums of the item detail lines may not equal the report total.

This surprising situation is caused by two factors; the method used to update the inventory record for receipts and the method used to print reports.

When an inventory item is replenished, the original cost and selling prices are replaced with the new figures. If the new figures are the same or the balance on hand is zero, there is no problem. However, should there be a difference and a balance of inventory remains unsold, the old inventory will be priced at the new cost and retail. Since the carry forward total has been adjusted by the increment only, it will now be different than the sum of the details.

When a sale occurs, the cost is calculated based on the main file cost figures available. If this is different than the purchase cost, the bottom totals will be distorted even more. It is quite possible that the ICS owner will not be aware that this is happening. When a total inventory report is requested, the system does not recalculate the totals. The source of the total data is the carry forward record, each detail line is only printed.

Aside from ICS distortions, there are also accounting drawbacks. Since the cost of the original item is lost, we can no longer use LIFO, FIFO and specific identification methods of valuation. All these methods require more knowledge about the composition of the inventory than is available from ICS. In fact, the only feasible method is average cost, however, to use this method, the program has to be modified. If you plan to use ICS for inventory valuation purposes, please be careful.

If you want to use ICS to establish a retail control to quantify shrinkage, please be aware that the distortions to the cost field also effect the retail field. Shortage measurement might be difficult under these circumstances.

Since ICS does not develop the extended total of the retail and cost field, I am including the following short program to develop the missing data. To use it, simply replace the password extension with your system's password. Hitting RUN and entering the number of items in the system gives you the extended cost.

Although this review is quite critical of ICS, this should not dissuade you from considering it. If you can live within its constraints and are unafraid to face some programming, it is a fine piece of work. It is easy to use, well-documented, and has worked well without strange bugs.■

```
10 INPUT"FILE SIZE TO BE TESTED";X
20 IF X=0THEN 10
30 GOSUB2530
40 FOR J=1TO X
50 GOSUB2540
55 GET2,JR
60 REM CONVERT FIELDS TO QUANTITY & AMOUNTS
70 QX=CVI(FQ$)
     AM=CVS(FC$)'SET FOR COST USE FP$ FOR
     RETAIL
80 PRINTFD$,QX,AM
     TC=TC+(QX*AM)
100 NEXT
110 PRINTTC
120 CLOSE
125 STOP
2530 OPEN"R",2,"DATAFILE PASSWORD"
     RETURN
2540 JR=INT((J-1)/4)+1
     JD=J-4*INT((J-1)/4)-1
2550 FIELD2,JD*62RSD$,19RSFD$,2RSFD$,4RSFC$,
     4RSFP$,10RSFL$,2RSFR$,2RSFR$,2RSFD$,
     8RSFM$,6RSFV$,2RSP1$,2RSP2$
RETURN
```

CAPTAIN 80

by Bob Liddil

Here's Captain 80 sitting in a fresh superhero uniform amidst programs stacked to the ceiling. Normally I would be in my disguise (as a mild mannered program reviewer), but the influx of GAME programs is so great that superhuman strength, speed and dexterity is required just to keep up.

As I said in my last column, I believe very strongly in games. I don't mean to arouse the game vs simulation controversy, but I want to illustrate my point. Looking at the current market of game software, I wonder what happened to imagination.

Not to say that there is no imaginative software on the market. Adventure, Dungeonquest, Sargon, Santa Paravia En Fumicio and a small list of others, are up, running and for sale. No, I'm talking about Star Trek version 2437, Blackjack version 2340.91, Hangman #9000 and all the countless variations of the programs loaded into the phone company's long distance phone lines in 1964.

I'd like to see some REAL simulations.....

How about a Life game based on the survival rate of apartment dwellers in New York City? You could call it Survive!! and program random degrees of Kung Fu ability into the potential victims. Or instead of racing forever around the Indianapolis Speedway, how about a spirited game of Freeway, simulating the San Diego Freeway south bound from LA—complete with smog, traffic jams and Eric Estrada on a Kawasaki, its blue light flashing.

I'm Not Kidding!

These ideas are just examples of possible themes compared to the incredible drivel that manages to work its way onto the marketplace. If I have to shoot down one more X-wing fighter, or hear Darth Vader's name again, I think I'll mail two pounds of magnetized iron to the producer in hopes of erasing his entire supply.

It is for better ideas in programming, that I announce the First Annual Captain 80 Program in a Paragraph Contest. Just sit down at your typewriter, desk, kitchen table or whatever, and write out in 50 words or less, a program you would like to see. It isn't necessary to actually write the program, just the idea.

Entries should be typed, double spaced and sent to Captain 80, c/o *80 Microcomputing*, Peterborough, New Hampshire 03458. Everyone who sends in an idea will receive a free membership in the SPPPp (Society for the Perpetuation of Perfect Programs Purveyed publicly). The best idea, as ajudicated by yours truly, will earn its creator $25. Send a stamped envelope to the above address and I'll send you a complete set of rules. Contest ends April 1st.

The lack of high quality games leads me to another sore spot. Where are all the educational programs? Hey, all you software producers out there: The TRS-80 is an educational tool! There are thousands of kids all over the world, playing Battlestar Galactica because you and your programmers have ignored geography, spelling, science, geology and all the other things that these active youngsters could be tearing into with a computer. If you are marketing, plan to market or have for potential sale, anything that resembles a kid-level educational program, send a copy to me, care of *80 Microcomputing* and I'll personally review it. Send it on tape or disc, with company name and where it can be bought.

Let me remind you, I'll review software—old and new—as it appears on the market or as it crosses my desk. I'll try (as only a true knight of computer justice can), to give a fair and honest opinion of the programs submitted by companies and individuals for my evaluation. You're invited to participate. Submit a short review of a program that you like or dislike.

And look: If you disagree with anything in the column, write about it!!! I'll try to answer each letter (in the beginning at least), personally.■

UNLIMITED 80's

by Sherry Smythe

Here's a success story about a man with no electronic or computer background, who saved his business with a TRS-80.

Bill Garlic and his wife, Priscilla, started Eastern States Traffic Service 25 years ago in a trailer with a $25 used IBM typewriter, a spirit duplicator and three customers. That first year was tough; they grossed only $750.

Eastern States Traffic publishes a book of tariffs that is a shippers only source of current freight rates for the country. Freight rate increases are proposed by the Tariff Bureau and either accepted or rejected by the ICC.

In 1955 there were five or six rate territories. Today there are about 35 (plus 10 intrastate). According to Garlic, rate changes occurred once or twice a year until 1979 when there were six.

Bill and Priscilla updated 300 to 400 pages of rates, each with 450 entries, by entering the percentage increase in a calculator and then typing the new rate on a tabulated form. It took two to three months to revise an old list. By kerchunking on the calculator at breakneck speed, the Garlics would still be two rate changes behind at the end of the year. It was obvious to Bill that something had to be done—and fast.

Finding Help

Last April, Bill picked up a pamphlet on the TRS-80 at his local Radio Shack. But trying to find further information about the 80 proved frustrating.

Bill says there are two kinds of computerists: Those who want to know everything about a computer; and his kind, who just want to solve a problem and then leave the computer to do its work. After many fruitless attempts, he finally ran across a Radio Shack Computer Center staffed with more sympathetic personnel and purchased a TRS-80. But still, Bill was short of software solutions for his business work.

By this time the walls were closing in on him. Rate changes were occurring faster than they could be updated. Bill figured if he was ever going to get the programming job done, he'd better do it himself. With a Level II and a handbook he started on the road to recovery—laboring hours getting the TRS-80 to run figures up and then down.

Finally, after putting a lot of faith and time into the project, he had a program that printed the updated figures on the screen a page at a time while Priscilla copied them. His disk drives were on order for four months when he purchased re-worked ones (for the regular price) entailing even more study of the disk manual.

Bill had an extra IBM Selectric II in his office which he found out could be interfaced as a printer. After much run around and many phone calls, an IBM rep directed him to a company in California with a Selectric adapter kit that would not jeopardize his service agreement with IBM.

Bill Garlic

Bill Garlic is a grandfather and a man who hates tinkering. But he purchased a soldering iron (he'd never put two wires together before), and spent a frustrating week installing the microcomputer interface.

After some false starts the Selectric started clacking away. Success! Bill had his first good night's sleep in weeks.

Reese Fowler of Instant Software and I visited Bill in his office in a lovely old New England hilltop home by the ocean. Despite the chilly day the pot-bellied, coal-burning stoves radiated warmth. Tucked away in the corner was the TRS-80 with its 32K, dual disk drives, and Selectric printer just waiting for the ICC to raise its rates again.

And next time maybe it will update those rates a little faster, because Reese helped Bill with some new programming routines. When I could pry him away from the computer Reese put on his photographer's hat and took the pictures for this column.

Bill says what took Eastern States Traffic two to three months, can now be done in two or three days. Increases that effect only parts of the country can be updated and in the mail in hours. ■

80 INPUT

From page 13

Appending Programs

I would like to comment on an article in your first issue of *80 Microcomputing* entitled NEW Restored, by Ken Fordham.

As I understand things, addresses 16633(LSB) and 16634(MSB) are the end of program pointer, not the next line pointer. This is helpful when you wish to append two or more programs; but first a note about the 0's that appear between each line number, and the 0 0 0 4 that appears at the end of a program.

The 0's that delineate one line from the next will be 1's above 32767, and the end of the program (which looks like 0 0 0 4) breaks down into this: The first 0 is the normal 0 that the computer puts at the end of a line, the next two 0's are a 'This is the end of the program' code, and the 4 means that the 0's preceding are single precision (if they were double precision it would be an 8).

When you key in NEW, you lose the end of program pointer, the beginning of program pointer (16548(LSB) and 16549(MSB)), and the first four bytes of the program become 0 0 0 4.

To append two programs, you PEEK the end of the first program (16633(LSB) and 16634(MSB)), POKE the beginning of program pointer to the end of the program pointer −2. (The reason you POKE it to the end −2 is because you want one 0 between lines, but don't want the additional two 0's that signal the end of a program.) Load the second program, and re-POKE the beginning of program pointer to the beginning of the first program (which will be different if disk is up, and will vary from one DOS version to another—so PEEK first if you use disk).

Continued to next page

To recover a NEWed program, you must re-POKE the end of program pointer + 2 (the reason it is + 2 is because you want the end or program code 0's at the end of a program), the first four bytes and if you use disk, the beginning of program pointer.

Some precautions: You must remember that you are not working in base 10; The largest number you may have is 256. It is equally important to realize that after you key in NEW, any POKEs you make, are going to write into the resident program.

That is, if you key in POKE 17129,241: POKE 17130,66:POKE 17131,0:POKE 17132,0:POKE 16633,0:POKE 16634,67 you will be (starting at 17129) overwriting the first 57 bytes of your program!

The way around this is to make the first two lines of your programs (lines 0 and 1) shift 7 REM'S. Since a' takes up three bytes, this gives you 16 bytes which will always be the same set of numbers in each and every program.

If you make only one POKE at a time (pressing enter after each entry), you cannot overwrite more than 10 bytes. To correct these 10 bytes the *easy* way just key in line 0' and line 1'. For increased protection, I recommend making the first three lines shift 7 REM's.

To find the end of the program if you forgot to PEEK prior to keying NEW, just key (in the DIRECT mode) FOR N=17129 TO 30000: PRINT PEEK (N);:NEXT

Hit enter and be ready to press SHIFT@. Look at the stream of numbers until you spot the first set of 0 0 0 4's—the end of a program. Look back one line so that you can calculate the address of the LSB of the pointer of the last line, add the number of bytes in the last line and add two to that number.

That will be the address of the end of the program to be POKEd into 16633 and 16634.

One last note: Table 1 is a ROM codes table. Missing from this table is code 251, which is ' (used in shift 7 REM). A shift 7 REM is 58 147 251. Also, code 188 which is listed as "TAB" is actually "TAB(".

Knowing the ROM codes allows you to change any command to any command (such as PRINT to LPRINT) with the following program (keyed in the DIRECT mode):

```
FOR N=17128 TO 32766:IF PEEK(N)=0 THEN N=(N+4):
NEXT N:ELSE IF PEEK(N)=0 AND PEEK(N+1)=0 AND
PEEK(N+2)=0 THEN END:ELSE IF PEEK(N)=X THEN
POKEN,Y:NEXT:ELSE NEXT
```

Where X = the old ROM code number, and Y = the new ROM code number. (Program by Craig Werner)

Mr. Robio L. Salmansohn
1855 Woodland Road
Abington, PA

Don't Give Me Grief

In at least two articles you printed an error that will cause many users much grief and OM errors! The Radio Shack EDTASM editor/assembler lists the 1A19H BASIC 2 entry point for a Ready. This address also gives OM errors and unsettles the BASIC Interpreter. The 0072H entry point works properly and will not cause any problems.

I have been using it in the RETURN to BASIC from our BEEP program (the return after the beep has been loaded and boots itself into the USR memory location, not the subroutine return).

Below you will find a copy of the letter I received from Tandy.

Harvey A. Kurtz Jr.
President K M C S
Box 02205
Cleveland, OH

There is a far better entry point to use instead of 1A19H and it is 0072H. A jump to this address returns a READY and does not give the OM error. If you should have any further questions, please contact us at our toll-free number.

Radio Shack Computer Services
Jahn I. Snodgrass, Jr.
Manager

Please for novices like myself have some short articles written in plain English.

Fewer Assumptions

I have just received my first copy of your magazine.

You stated that you wanted feedback about your first issue. As a medical school professor, I am using my three disk drive 48K TRS-80 for word processing, writing of seminars and student objectives, class design and short entertainment programs.

I am interested in learning more about the inner workings of the TRS-80, especially machine and assembly language. However I find the articles in your first issue way over my head. Most of them discuss machine language methods to do things with or to the TRS-80. Unfortunately the jargon and jumps of language are indecipherable to me. Your writers obviously assume everyone is as sophisticated as they are. Some introductory articles on how to use machine/assembly language would be valuable. Some introductory articles on how to function without cassette tape loading would be helpful. Some introductory articles on jargon would be helpful. What is a hash? Some introductory articles on how to get at the contents of ROM and RAM and what to do with the data when it shows up on the video screen would be very useful.

Additionally the problems of circuit diagrams make little sense to me and probably to most people who want to use the computer to help communicate with others and write programs that will run faster.

I know that some of this information is in the Radio Shack hardware books, but even there the information is cursory.

In the article on decision-making there were a number of errors that made the program breakdown as written. I changed line numbers and instructions somewhat.

Please for novices like myself have some short articles in plain English, that if accumulated and placed in a ring binder will, over the course of several months, explain how to get the most out of my computer. How do you use the editor/assembler, for example (the Radio Shack instructions are awful!). How do you use machine language?

I understand the theory of computers. I have been working with systems analysts analyzing mass data and clinical activities for years. I have a moderate understanding of programming in BASIC and some acquaintance with FORTRAN but the inner works, and how to access them and use them, escape me at present.

Christopher M. G. Buttery MO MPH.
4609 Templar Drive
Portsmouth, VA

Useful Locations

A long time ago, during 1977, I had just gotten started in computers. I went to the computer store and bought some magazines and books, eager to learn more about this fascinating subject. One of the magazines I bought was *Kilobaud*. When I got home, I browsed through everything. The *Kilobaud* magazine was sort of thin and the table of contents was on the front cover. I hated the typeface, because all the other computer magazines had nice neat ones. So, I got turned off to it.

Then, in November of 1979, I was at the computer store again and I picked up an issue of *Kilobaud*. It looked really nice, was thicker and the articles looked interesting. I bought it. After I had gotten a chance to read it, I realized that this was a very good magazine and from reading the subscription ad, I knew I missed out on a lot of good articles. So I went and got a subscription to *Kilobaud* and *80 Microcomputing*.

In regard to "Hidden codes and Missing chips," I discovered some other memory locations that might be of use. The big one is x40B1, which is the pointer to the end of memory minus two. If you have 16K and you haven't reserved any memory, then PEEK(16561) should be 254 and PEEK(16562) should be 127.

A nice way of altering this would be to POKE the new values in those locations, then execute a CLEAR. This is a way to change the memory size without losing your program or reinitializing the computer. 16561 is the low order byte and 16562 is the high order byte. In 16K, the 127 and 254 from above mean 127*256 + 254, which is 32766, plus two is 32768, which is the memory size used when no memory is reserved.

Jim Raden
602 W. Wayne
Maumee, OH

80 NEWS

edited by Michael Comendul

A Computer for Business

How does a small engineering company that needs—but can't afford—a large computer, survive in today's business world with larger, computer-equipped competitors?

By adapting the hobbyist's microcomputer.

The John West Company, specializing in petrochemical and refinery-related engineering services, did just that—using the $2500 Radio Shack TRS-80 as a base. The entire system cost under $5000, interfacing the company's IBM Selectric II typewriter as a printer.

"We couldn't afford the large computers our competitors utilized, but we felt like we had to take advantage of the efficiencies, and just plain better job, that a computer affords in special engineering applications," recalled West.

While no computer on the market could do what the company wanted at an affordable price, several small microcomputers came close. The hardware was adequate, but software was limited mostly to BASIC programs.

Available maintenance and a national reputation prompted the selection of a Radio Shack TRS-80 computer as the base of the company's computer system.

West's 48K TRS-80 has two floppy disks. The first handles 57,000 bytes of memory; the second handles 80,000 bytes.

"With standard software (an Electric Pencil program and general business programs), we have basically the same word processing capability provided by much larger machines—at a very modest price," West explained.

Also standard are general business programs that cover everything from accounting and billing to payroll.

Additional engineering programs, often unavailable from computer manufacturers, were specially developed and usually inexpensive.

For less than the cost of a part-time employee John West has a computer system ideal for his small, technically oriented business.

John West: His entire system cost under $5,000.

Hamboree/Computerfest

A Hamboree and Computerfest will be held on Sunday, March 30, at the Maryland State Fairgrounds at Timonium, Maryland.

In addition to commercial exhibitors including Radio Shack and Heath, a number of smaller firms will be selling software and accessories. Several computer stores will also be exhibiting. The fest is planned around an equipment flea market.

Speakers scheduled include Wayne Green, publisher of *73 Magazine, Microcomputing* and *80 Microcomputing*. One of Green's two public appearances in 1980, he will be speaking about marketing ideas in microcomputing.

For more information contact Joe Lochte, 2136 Pine Valley Drive, Timonium, MD 21093.

Children's Program

Computers are for kids—at least that's what the 30 students at the Woodland School in Spotswood, New Jersey believe.

Students in grades four through seven, have been writing their own programs covering everything from geography to graphics on a Radio Shack TRS-80 microcomputer for about a year now. They are part of an advanced student program that obtained a TRS-80 in January of 1979 with a grant from the New Jersey Department of Education.

Now at the request of teachers within the school, the students are creating and writing computer programs for use in the classroom. The programs are designed for grades one through seven in mathematics, social studies, science and language arts and to prepare student for quizzes.

The students at Woodland School were filmed recently for the syndicated children's television program, *Kidworld*, a children's program designed to give youngsters an opportunity to report what is going on in their world. All ideas for the content of the program are submitted by the youngsters themselves.

According to their instructor, Laura Zatz, "The TRS-80 represents a challenge to my students because it is something new in learning and promotes creative and logical thinking. Even slow learners can benefit from using the TRS-80."

Software Catalog for Model-II

A software catalog for Radio Shack's TRS-80 Model-II computer systems describing accounts receivable, accounts payable, general ledger, payroll, inventory, rental management, order entry and a variety of financial and mathematical programs is available from National Marketing, Inc., Hollywood, FL.

The programs operate on a 64K Model II with built-in disk. They are priced from $15 to $100.

The catalog is offered free.

Readers Service ⌐170

Index Sequential Access Method

An Index Sequential Access Method for controlling business application files on diskette is available from Johnson Associates, Redding, CA.

The ISAM system is a series of subroutines the user includes in his program. Calls to these subroutines store or retrieve data by referencing a key field within the record. An additional set of utility programs allows the user to create a new data file or to reorganize an old one.

All ISAM files are supervised by the TRS-80 Disk Operating System, thereby providing standard space allocation, directory, copy, kill, backup and password services.

Any record field can be designated as the key field and all subsequent adds and retrieves are based on the content of this field. Records can be added, updated or deleted at any time and in any sequence.

The system allows up to 15 ISAM files to be open simultaneously, however, memory requirements for such an application would be large.

Readers Service ✓174

Control Your Peripherals

The TRS-80 Breadboard, a hardware device, available from Group Technology, Ltd., Check, VA, allows the microcomputer user to design custom interfaces for his peripherals. The TRS-80 Breadboard contains bi-directional data bus buffers, a logic probe, a solderless breadboard, and an eight-device address decoder that can be used for either accumulator I/O or memory-mapped I/O. The user can select not only the mode of operation for the device address decoder, but also the four or twelve most significant bits of the device address.

The Breadboard allows the user to communicate with control signals in the microcomputer not readily accessible. It can be used with 4K Level II TRS-80's up to dual floppy disk 48K Level II systems.

A 190-page textbook by Dr. Jon Titus, *TRS-80 Interfacing*, instructs the Breadboard user in the construction of device address decoders, input ports, output ports and synchronization signals. Hardware interfaces and software listings are shown for A/D and D/A converters, programmable interface chips, data loggers, a traffic light controller and a digital logic tester.

The text includes 18 experiments that can be performed by the user with expected results. All programming is done in BASIC.

The TRS-80 Breadboard is available as a parts list and instructions for $3.00 or as a kit. Kit prices range from $25 to $250.00. *TRS-80 Interfacing, Book 1*, is priced at $8.95, plus $1.00 shipping and handling.

Readers Service ✓177

TRS-80 Breadboard

Level II Guidebook

Dr. David A. Lien, author of the *TRS-80 User's Manual*, has released *Learning Level II*, a fully illustrated guidebook created specifically for users of the Level II TRS-80.

The book, directed toward the novice, examines all Level II BASIC beyond Level I with step by step approaches covering special characteristics.

The manual explains how to use the editor, dual cassette operations, the expansion interface box with the real-time clock, printers and other peripherals.

Learning Level II costs $15.95 plus $1.45 postage and handling and is available from Compusoft, San Diego, CA.

Readers Service ✓178

Checkbook Without Tape Record-keeping

Manhattan Software, New York, NY, has released its latest program, Checkbook Plus, which provides a once-a-month solution for checkbook and bank-statement reconciliation.

The user enters his checkbook balance, the bank's balance, outstanding checks and bank charges to check his own balance against the bank's figures. A special arithmetic-checking section with optional automatic per-check-charge insertion verifies each intermediate balance.

If figures don't agree, the program guides you through possible error sources.

The cost is $9.95.

Readers Service ✓179

Custom Furniture

Custom wood office furniture providing maximum work surface with accessibility, is available for the TRS-80 microcomputer system from AVS, Alviso, CA.

The unit fits into the corner and mates with an optional printer/typewriter platform or storage hutch.

All TRS-80 units, though built-in, simply drop into place and do not require any mounting hardware or tools.

Custom TRS-80 cabinets

The standard unit holds the monitor, cassette, keyboard and expansion interface. Options are available for mounting the screen printer and/or disk drives.

Readers Service ✓183

TRS-80 Microwave

Interactive Microware, Inc., State College, PA, is developing a library of Radio Shack TRS-80 software. These include the following: Basex Compiler, an easy-to-learn language that runs up to 20 times faster than BASIC ($25 + $8 for 97-page manual); Mirrorays, a game in which the user flashes rays of light into a black box to locate hidden mirrors, which light up and reflect the rays when hit; a Compact Graphics Interpreter creates graphic designs with a simple set of numbers; a Lunar Lander Simulator provides real-time simulation and control of a lunar module; Battlegrid, a real-time game enabling two players to attack each other's forces. The number, type and size of battle pieces can be specified by the players.

All of these programs operate on a 16K Level II TRS-80 and sell for $7.95, except as noted.

Readers Service ✓165

BASIC Protection

Data Associates, Framingham, MA, has released a program, Unlist8, that will automatically protect BASIC programs against unauthorized modification.

It runs on a single disk system with 32K memory and inserts hidden passwords and copyright notices selected by the user so the program cannot be listed or printed though it can still be RUN, CSAVEd, CLOADed, disk loaded and disk saved as usual.

Options permit unlisting all lines, or each n'th line, or specified blocks of line numbers. This program can also be used to relist a protected program provided that the password is known. It can relist each line, or blocks of specified line numbers.

✓ Reader Service—see page 147

Unlist8 is provided with an instruction manual and three copies on cassette for $19.95 postpaid.

Readers Service ✓166

The Pencil/Pal

MicroComputer Specialists, Elkins Park, PA, has released Pencil/Pal to be used in conjunction with the Electric Pencil.

With Pencil/Pal you can automatically merge your letters with an address file and LPRINT them.

Pencil/Pal is compatible with the lowercase modification. One or two fields within your address file, even area codes or zips, may be used to select letters to be printed.

The program costs $35. Send $5 for documentation only—deducted from purchase price.

Readers Service ✓180

Storage and Retrieval for TRSDOS

ISAR is a BASIC data base management system that uses TRSDOS random file structures and the limited TRS-80 chaining techniques. This means you only have as much of a program in memory as necessary.

ISAR consists of six modules that create any number of new files, define all elements within each file and manipulate them according to a menu. Files are sorted with Shell-Metzer.

ISAR is available on cassette for $13.95 or diskette for $16.95 from The Alternate Source, Lansing, MI.

Readers Service ✓181

Income Tax For The TRS-80

This book contains more than 40 1979 Income Tax programs in 100 pages for the TRS-80.

Most of the programs are for LPRINT and several show how to convert these to PRINT only.

Programs cover child care, personal residence, special 10-year averaging and underpayment. A chapter is devoted to tax credits.

The price is $14.95 and is available from Gooth Software, Louis, MO.

Readers Service ✓182

Clock Modification

Mumford Micro Systems, Summerland, CA, has released a new clock modification for the TRS-80. The SK-2 3-Speed Mod is a small circuit board with five integrated circuits which can be mounted inside the keyboard unit or externally.

It interrupts the main clock line to the Z-80 and allows switching from normal speed to a 50 percent increase and a 50 percent decrease. Switching is controlled by a toggle or by software.

Disk users can add a control line to the expansion interface to automatically force a return to normal speed at any time. This eliminates the need to write speed commands into your programs or modify the operating system.

An LED indicates when the computer is not at normal speed.

The SK-2 comes assembled and ready to install for $24.95 (plus .75 postage). Only four connections are necessary to the computer.

Readers Service ✓173

Business-Aides

Occupational Computing Company, Inc., Woodland Hills, CA, has released its business aides system.

These accounting and management programs include: Accounts Receivable, Billing and Inventory Control for both manufacturing and finished goods; Accounts Payable; Payroll; Client Accounting.

Prices range from $350 to $1495.

Readers Service ✓168

Color Display from Percom

The Electric Crayon, a computer-operated color graphics generator/controller from Percom Data, Garland, TX, is designed to generate color displays on either a TV set or monitor.

The Electric Crayon includes its own ROM operating system, EGOS, that accepts single-character commands directly from a parallel ASCII keyboard or program-generated commands.

A self-contained control computer, the Electric Crayon provides 1K-byte of on-board program RAM, an EPROM chip and a second dual bidirectional 8-bit port for peripherals. It has 10 display modes—an alphanumeric-semigraphics mode, a high-density semigraphics mode and eight graphics modes. Up to eight colors can be generated.

The Electric Crayon measures 2-½ inches high by 9 inches deep by 12 inches wide and sells for $249.95.

Other options include:

BASIC language color graphics programs on minidiskette.

A 34-conductor ribbon cable to interconnect the Electric Crayon to the TRS-80 printer port.

RAM chips for adding refresh memory for higher density graphic modes.

Readers Service ✓169

Percom Electric Crayon

Machine Language Sorts

A machine language Generalized Subroutine Facility for the Model II is available from RACET, Orange, CA. Its functions include multi-key/multivariable in-memory sort, multi-key character string in-memory sort, USR PEEK and POKE capability—both byte and word and fetch argument. The subroutine will compress and uncompress data, move it in blocks and propogate across arrays.

The subroutine will sort 1000 elements in six seconds and carries up to 15 arrays together with multiple mixed ascending/descending keys.

The cost is $50 on your DOS diskette.

Readers Service ✓172

Payroll Program

Small businesses with TRS-80 Model I Computers can now utilize Data Train, Inc's Payroll for their dual mini-disk 32K systems.

The DTI Payroll allows 50 employees per diskette and runs in all states with state, federal, and local taxes and employee records set by the user. Other features include: Monthly, quarterly, yearly pay and hour records; recording of handwritten (after the fact) checks; departmental reporting; maintains W-2 and 941; special reports for departments, unions, earnings, tax.

Fixed programmed reports include: Checks and/or stubs, register, journal, employee list/records, mailing labels and others.

The price is $235 and is available from Data Train, Inc., Grants Pass, OR.

Readers Service ✓175

Power Line Filter

To eliminate most of the sensitivity of the TRS-80 to power line noise and reduce its television interference, Percom Data Company, Garland, TX, has introduced a simple power line filter. The following materials are available from Percom and (except for the filter) most hardware and electronic stores:

Corcom 10R3 EMI Filter
3-wire power cord (Belden #17237B)
Power cord strain relief (H.H. Smith #939)
117 V ac Socket (H.H. Smith 1280-103)
4" x 2¼" x 2¼" Minibox (Bud CU-2103-B)
6-32 machine screw and hex nuts

Readers Service ✓163

DISCOVER THE MAGIC OF WORDPROCESSING AND TURN YOUR **TRS-80*** INTO A VERITABLE **"OFFICE WIZARD"** WITH A **"NEC SPINWRITER"**

NEC 5530

(CENTRONICS I/O)

What makes NEC your best typewriter quality printer value?

1. The one and only print **"thimble."** SPINWRITER'S DURABLE (up to 30 million impressions) and unique reinforced plastic print element. Up to 128 characters per print element, with a wide variety of typefaces to choose from.

2. **SPEED and RELIABILITY.** The SPINWRITER gives you up to three times the speed of IBM "Golf Ball" type mechanical printers, along with the advantages of a true electronic output printing device.

3. **HUMAN EAR COMPATABILITY.** The low pitched strumming noise made by the NEC Spinwriter won't leave your ears and nerves aching after a hard day's use, as Dot Matrix and mechanical printers tend to.

THE NEC SPINWRITER
IS FULLY SERVICED BY NEECO.

NEC 5530 SPINWRITER (plug compatible to TRS-80*) $2495.00
Word Processing Software "Electric Pencil" ..... $ 100.00

Call or Write today for NEC Specifications Sheet. (Specify TRS-80*)

***TRS-80 is a Trademark of Tandy Corp.**

✓143

NEECO
679 Highland Ave.
Needham, MA
02194

Mon-Fri 9:30-5:30
MasterCharge &
Visa Accepted

(617) 449-1760

Instant printing/instant pricing, use a computer to impress customers—and give accurate quotes.

Printer's Apprentice

Richard Barnes
1515 S. Glendale
Sioux Falls, S.D. 57105

One small-business application where the microcomputer has proved effective and cost efficient is the print industry. There has been an astounding revolution in this field prompted by the development of the "instant printing" or "quick print" business.

According to *Printing Impressions*, an industrial trade journal of the printing world, the quick print business has made commercial printing available to many businesses and individuals who had previously been using office mimeographs and copying machines. The magazine also pointed out that the average quick print plant in the United States last year grossed an average of $174,000.

A new problem emerged with the development of the quick print plant and it is a problem that can be resolved through the use of the microcomputer.

Variable Pricing

While each small print shop has its counter price sheet, giving customers prices for jobs that require 100 or 500 copies, the small office staff is often perplexed when asked to quote a price that is not on the limited price sheet. It is an area where mistakes can be costly, and a small printer can find that he has made a commitment on a bid while omitting an important cost element.

One small printer recently submitted a bid on a difficult printing job which was $400 under the next highest bid. He won the bid, and realized that he had forgotten to include his printing costs for the second side.

If the small printer grosses the estimated $174,000 a year, with a one percent error factor he could easily lose $2,000 a year—enough to pay for a small computer system.

The program for a small quick print shop can be relatively simple. It includes instant access to a price schedule. By writing a program that asks for all data relating to the printing costs, error by omission is next to impossible.

Secondly, such a program offers the public a fair print pricing policy.

As a good businessman, the printer should know his fixed overhead, how many copies he can produce in one hour and his paper costs. The only other variable should be the number of copies needed. With computerized printing prices, the customer is assured that the price given is accurate and fair.

The instant printer tries to turn around his printing business within one day or two, at the most. Instant printing, necessitates instant pricing.

The Program

Our pricing program is designed for the convenience and use of the walk-in customer. It can also be used by the staff in quoting prices by phone, but the public is encouraged to use the system.

Since the unit sits unattended in our lobby, a routine is used to lock out inappropriate responses throughout the program. GOTO 1000 clears the screen and sends the program to an endless loop, telling the user to call someone for assistance. Few know how to BREAK out of the loop and the computer keyboard remains locked so that the user cannot toy with the machine while waiting for a clerk.

The two most important variables in determining the price of a printing job are the number of copies, and the size and quality of the paper stock being used. It is also important that the businessman determine his overhead

and the cost per thousand of printing a job.

Line 55 sets the press run (PR) at $7 per thousand copies for overhead. That figure varies with each shop depending upon staff, rent and output.

The program begins by asking the user what kind of print bid he needs. Line 210 sends the program to the routine selected by the user.

The most common selection is sheet fed printing. To determine the price, the program needs the number of copies, the size of paper, whether it is to be printed on one side or two and whether a surcharge for color stock (paper) should be added.

Using a minimum order of 100 copies, line 310 takes the value of Q (quantity) and converts it to a minimum of 100 copies for the purpose of billing. Because different stocks will vary according to size, lines 390 through 410 determine the paper (PA) costs. If a printer adds a percentage to his actual costs, this is the place to do it so that it is included in the final billing. If your paper costs go up, a simple edit of these lines adjusts the program to the increase.

GOSUB 2000 asks whether the order is to be printed on one side or two. If two sides are needed, a second plate charge is added (Line 2030), the press run (PR) is doubled (Line 2040)

Our pricing program is designed for the convenience and use of the walk-in customer.

and a 10 per cent charge is added to the paper costs for waste and handling.

GOSUB 3000 determines whether the printing job will be run on 20 lb. white or 20 lb. color stock. The additional markup in GOSUB 3000 for color paper by size, seems to be an industry standard. The program adds 50¢ per hundred for 8½ x 11 and adds 75¢ per hundred for 8½ x 14. This may vary with market areas and should be checked locally.

Determining Price

The main algorithm in line 470 figures the cost (C) as the value of the plate (variable W at $2 each), plus the press run (PR) times M (the number of copies converted to a thousand or a fraction thereof), plus the paper costs (PA) times M, plus the surcharge for color stock.

The program then asks for bindery services in lines 510-530. Folding, stapling, collating, padding, cutting and drilling are all included in the offer and the user selects the service needed.

On occasion, the customer needs his order folded, collated and stapled. The program offers the customer multiple bindery services by looping through the selection until the customer enters a response to let the computer know that no other services are needed. The

Program Listing.

```
55 PR=7
60 PRINT "              HELLO.  I AM A TRS-80 COMPUTER"
70 PRINT
80 PRINT "I CAN ANSWER MOST OF YOUR QUESTIONS ABOUT PRINTING PRICES"
90 PRINT "JUST FOLLOW THE INSTRUCTIONS, ENTERING THE PROPER RESPONSE."
100 PRINT "BE SURE TO HIT THE 'ENTER' KEY AFTER COMPLETING EACH ANSWER."
110 PRINT: PRINT "FOR WHICH ITEM WOULD YOU LIKE A PRICE QUOTE?"
120 PRINT "             (1) OFFSET PRINTING -- SHEET FED"
130 PRINT "             (2) ENVELOPES"
140 PRINT "             (3) BUSINESS CARDS"
150 PRINT "             (4) CARBONLESS BUSINESS FORMS"
160 PRINT "             (5) TYPESETTING"
170 PRINT "             (6) GRAPHIC ARTS"
180 PRINT "             (7) COLOR PRINTING"
190 PRINT "             (8) OTHER..."
200 INPUT L: IF L=0 OR L>5 THEN 1000
210 ON L GOTO 250,4000,4500,5000,1000,1000,1000,1000
250 CLS
260 PRINT "    ALL PAPER IS ON 20 LB. WHITE BOND"
270 PRINT: PRINT "ANSWER BY ENTERING THE CORRECT RESPONSE AND HIT THE"
280 PRINT "       'ENTER' KEY."
290 PRINT: PRINT "HOW MANY COPIES DO YOU WANT?"
300 INPUT Q : IF Q=0 THEN 1000
310 IF Q<100 THEN Q=100
320 M=Q/1000
330 PRINT "      WHAT SIZE PAPER WOULD YOU LIKE?"
340 PRINT "            (1) 8.5 X 11  ?"
350 PRINT "            (2) 8.5 X 14  ?"
360 PRINT "            (3) 11  X 17  ?"
370 INPUT K
380 IF K=0 OR K>3 THEN 1000
390 IF K=1 THEN PA=6.52
400 IF K=2 THEN PA=8.30
410 IF K=3 THEN PA=13.05
450 GOSUB 2000
460 GOSUB 3000
470 C = W + ( PR * M ) + ( PA * M ) + R
480 C = INT((C+.005)*100)/100
490 CLS
500 PRINT@525, "YOUR PRINTING ORDER WILL COST $";C
510 PRINT: PRINT "     DO YOU NEED BINDERY SERVICES?"
520 PRINT "           (1) YES"
530 PRINT "           (2) NO "
540 INPUT P
550 IF P=1 THEN 600
560 PRINT "     THANK YOU AND HAVE A NICE DAY !"
570 PRINT:GOTO 930
600 CLS
610 PRINT "THE FOLLOWING SERVICES ARE AVAILABLE WITH YOUR"
620 PRINT "  PRINTING ORDER.  PLEASE ENTER THE NUMBER OF"
630 PRINT "  THE SERVICE YOU NEED."
640 PRINT:PRINT TAB(45) "MIN."; TAB(58) "COST"
650 PRINT TAB(45) "CHG."; TAB(57) "PER 100"
660 PRINT:PRINT "(1) FOLDING"; TAB(45) "$2.00";TAB(58) "$ .50"
670 PRINT "(2) STARLING OR SADDLE STITCHING";TAB(45) "$2.00"; TAB(58) "$1.00"
680 PRINT "(3) COLLATING COPIES";TAB(45) "$2.00";TAB(58) "$ .50"
690 PRINT "(4) PADDING - USUALLY 50 COPIES PER PAD";TAB(45) "$2.00";TAB(58) "$ .30"
700 PRINT "(5) CUTTING - PER 500 COPIES"; TAB(45) "$2.00";TAB(58) "$ .25"
710 PRINT "(6) DRILLING - PER HOLE";TAB(45) "$2.00";TAB(58) "$ .25"
720 PRINT "- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -"
730 INPUT G
740 ON G GOTO 750,760,770,780,790,800
750 F=.5  : GOTO 810
760 F=1 : GOTO 810
770 F=.5 : GOTO 810
780 F=.3 : GOTO 810
790 F=.2  :  GOTO 810
800 F=.25 : GOTO 810
810 B=(Q/100 * F)
820 IF B<2 THEN B=2
830 CLS
840 PRINT "YOUR PRINTING COSTS ARE  $";C
850 B=INT((B+.005)*100)/100
860 PRINT "YOUR BINDERY COSTS ARE   $";B
870 PRINT "YOUR TOTAL JOB COSTS     $";(C+B)
880 T=(C+D)*.05
890 T=INT((T+.005)*100)/100
900 PRINT "          SALES TAX:     $";T
910 U=C+D+T
920 PRINT:PRINT "             TOTAL:  $";U
930 PRINT:PRINT "IF YOU NEED ANOTHER PRINTING QUOTE HIT THE"
940 PRINT "          'ENTER' KEY."
950 INPUT I:CLS:GOTO 60
1000 CLS
1010 PRINT@330, "     PLEASE CALL SOMEONE FOR ASSISTANCE"
1020 FOR X=1 TO 500: NEXT X
1030 PRINT:PRINT "     MY MEMORY BANKS DO NOT HAVE THE INFORMATION NEEDED"
1040 FOR X=1 TO 500: NEXT X
1050 PRINT:PRINT "                    TO RESPOND TO YOUR REQUEST"
1060 FOR X=1 TO 750: NEXT X
1070 PRINT:PRINT "                           THANK YOU"
1080 GOTO 1000
2000 PRINT "DO YOU WANT IT PRINTED ON 1 SIDE OR 2 ?"
2010 INPUT S
2020 IF S=0 OR S>2 THEN 1000
2025 REM *** FIGURES THE COST OF THE PLATE AT $2 ***
2030 W= S*2
2040 IF S=2 THEN PR=(PR*2): IF S=2 THEN PA=PA+(PA*.1)
2050 RETURN
3000 PRINT "DO YOU WANT IT PRINTED ON COLORED PAPER?"
```

Continued on next page

Continued . . .

```
3010 PRINT "          (1) YES"
3020 PRINT "          (2) NO "
3030 INPUT R : IF R=0 OR R>2 THEN 1000
3040 IF R=2 THEN R=0: IF R=0 THEN 3000
3045 REM *** COMPUTES SURCHARGE FOR COLOR PAPER BY SIZE ***
3050 IF K=1 THEN R=(N*5)
3060 IF K=2 THEN R=(N*7.5)
3070 IF K=3 THEN R=(N*10)
3080 RETURN
4000 REM *** ENVELOPE PRICING ***
4010 CLS
4020 PRINT "        HOW MANY ENVELOPES DO YOU NEED?"
4025 PRINT "  (DUE TO PRESS SET UP, MINIMUM ORDER OF 500 PLEASE)"
4030 INPUT N : IF N=0 THEN 1000
4035 IF N<500 THEN N=500
4040 M=N/1000
4050 PRINT:PRINT "WHAT SIZE AND KIND OF ENVELOPE DO YOU NEED?"
4060 PRINT "              (1) SMALL 6 3/4 WHITE PLAIN"
4070 PRINT "              (2) SMALL 6 3/4 WHITE WINDON"
4080 PRINT "              (3) NO. 10 WHITE PLAIN"
4090 PRINT "              (4) NO. 10 WHITE WINDON"
4100 PRINT "              (5) NO. 10 COLORED STOCK"
4110 PRINT "              (6) OTHER..."
4120 INPUT B
4130 IF B=0  OR B>5 THEN 1000
4140 REM *** COMPUTES SURCHARGE FOR ENVELOPE STOCK ***
4150 IF B=1 THEN E=1.20
4160 IF B=2 THEN E=1.50
4170 IF B=3 THEN E=1.50
4180 IF B=4 THEN E=1.75
4190 IF B=5 THEN E=3.50
4200 PRINT:PRINT "DO YOU NEED TYPESETTING OR IS THE COPY CAMERA READY?"
4210 PRINT "           (1) TYPESETTING NEEDED"
4220 PRINT "           (2) CAMERA READY COPY "
4230 INPUT D
4240 IF D=2 THEN D=0
4250 IF D=1 THEN D=5
4260 C=2+(7*N)+(6.52*N)+((E*10)*M)+D
4270 C=INT((C+.005)*100)/100
4280 CLS:PRINT@515,"THE COST OF PRINTING YOUR ENVELOPES WILL BE $";C
4290 GOTO 930
4500 REM *** BUSINESS CARD ROUTINE ***
4510 CLS
4520 PRINT "                        BUSINESS CARDS "
4530 PRINT "   WE HAVE A WIDE SELECTION OF BUSINESS CARDS STARTING AT"
4540 PRINT "                        $ 8.90 FOR 500"
4550 PRINT "                  OR    $10.90 FOR 1000"
4560 PRINT "OUR STANDARD LINE COSTS $13.90 PER 1000 WITH THE ADDITIONAL"
4570 PRINT "CHARGES FOR THE FOLLOWING:"
4600 FOR X=1 TO 1000:NEXT X
4690 PRINT
4700 PRINT "        ART WORK CUTS.....................$2.00"
4710 PRINT "        PHOTOGRAPHS.......................10.00"
4720 PRINT "        PRINTING ON BACK SIDE ............10.00"
4730 PRINT "        SCREENS ..................PER SCREEN  5.00"
4740 PRINT "        BLEED-OFFS ...............PER SIDE.. 5.00"
4750 PRINT "        CLOSE REGISTRATION OF COLOR.......10.00"
4760 PRINT "        CAMERA REDUCTIONS ..................  4.50"
4770 PRINT "        VERTICAL LAYOUT ....................  2.00"
4775 FOR X=1 TO 5000:NEXT X
4780 PRINT:PRINT "PLEASE ASK TO SEE OUR SAMPLE BOOKS AND ASK SOMEONE"
4790 PRINT "                FOR ASSISTANCE."
4800 GOTO 930
5000 REM *** CARBONLESS BUSINESS FORMS ***
5010 CLS
5020 PRINT "HOW MANY FINISHED SETS DO YOU NEED?"
5030 INPUT F
5040 PRINT:PRINT "HOW MANY PARTS TO THE FORM?"
5050 PRINT "        (1) TWO PART FORM"
5060 PRINT "        (2) THREE PART FORM"
5070 PRINT "        (3) FOUR PART FORM"
5080 PRINT "        (4) FIVE PART FORM"
5090 INPUT P: P=P+1
5100 PRINT "DO YOU WANT PRINTING ON ONE SIDE OR TWO?"
5110 PRINT "          (1) ONE SIDE"
5120 PRINT "          (2) TWO SIDES"
5130 INPUT S:W=S*2
5140 PRINT:PRINT "WHAT IS THE FINISHED SIZE OF THE FORM?"
5150 PRINT "          (1) 4 1/2  X  5 1/2"
5160 PRINT "          (2) 5 1/2  X  8 1/2"
5170 PRINT "          (3) 8 1/2  X  11  "
5180 PRINT "          (4) OTHER..."
5190 INPUT D
5200 IF D=1 THEN E=4
5210 IF D=2 THEN E=2
5220 IF D=3 THEN E=1
5230 IF D=4 THEN 1000
5240 N=((F*P)/E)/1000
5250 C=W+((PA*N)*S)+(21.52*N)
5255 C=INT((C+.005)*100)/100
5260 CLS:PRINT@515,"THE COST OF PRINTING YOUR CARBONLESS FORMS IS $";C
5270 GOTO 930
```

program totals the amount based upon the number of copies handled and prints them in line 860.

Lines 830 through 920 display the total breakdown for printing and binding, including sales taxes. Line 880 adds 5 per cent sales tax and should be altered depending upon your state.

The routine for pricing envelopes follows the same format, while the business card routine displays basic prices suggested by (specialty) printing houses and suggests that the customer review the business card sample book—a most difficult task for a computer.

The routine that handles pricing for carbonless forms has been helpful in avoiding bidding errors. Carbonless business forms vary from two to five copies each. Thus, an order for 500 sets of a four-part form, really requires 2,000 impressions instead of 500. Often, the forms are run two-up and later cut apart. This means that the 500 sets of a four-part form, run 8½ x 11 but then cut to 5½ x 8½, only require 1,000 impressions. You can see how easily a job might be over or under estimated.

The computer routine handles all factors and has won us business when our competitors were confused and erred in putting their bids together.

After gathering all of the necessary information, the real number of copies needed (N) is determined by taking the total number of sets needed (F in Line 5030) times the number of parts to the form (P in Line 5090). Divide that number according to whether one, two or four forms will be run in the original (E in Lines 5200-5220) and convert it into thousands or a fraction thereof in Line 5240.

Line 5250 determines the cost (C) as the cost of the plate (W—adjusted in Line 5130 for one side or two), plus the press run (PR) times N, multiplied by the number of sides to be printed (S). Then add the cost of the carbonless stock times N, the number of copies.

Conclusion

This program allows the computer hobbyist to provide consultant services to local printers. While our own program has been greatly expanded to add a wider variety of paper, this program should be enough to get one started.

A word of caution: we have included a line in our program that offers more competitive bids on orders of more than 10,000 copies. The price greatly decreases on large orders and the program here cannot be competitive with large commercial printers. The quick printer might also want to bid under the computer price by giving color stock at cost in order to secure the winning bid.

How does it work in practice?

One customer asked for 200 copies of a flyer and quickly added that she really needed 225 but did not want to pay for 300. She was informed that she could enter any number into the TRS-80 and it would compute the price to the nearest fraction of a penny. She ordered and paid for 225 copies and said she was pleased she did not have to pay for 300 copies, "like I did down the street."

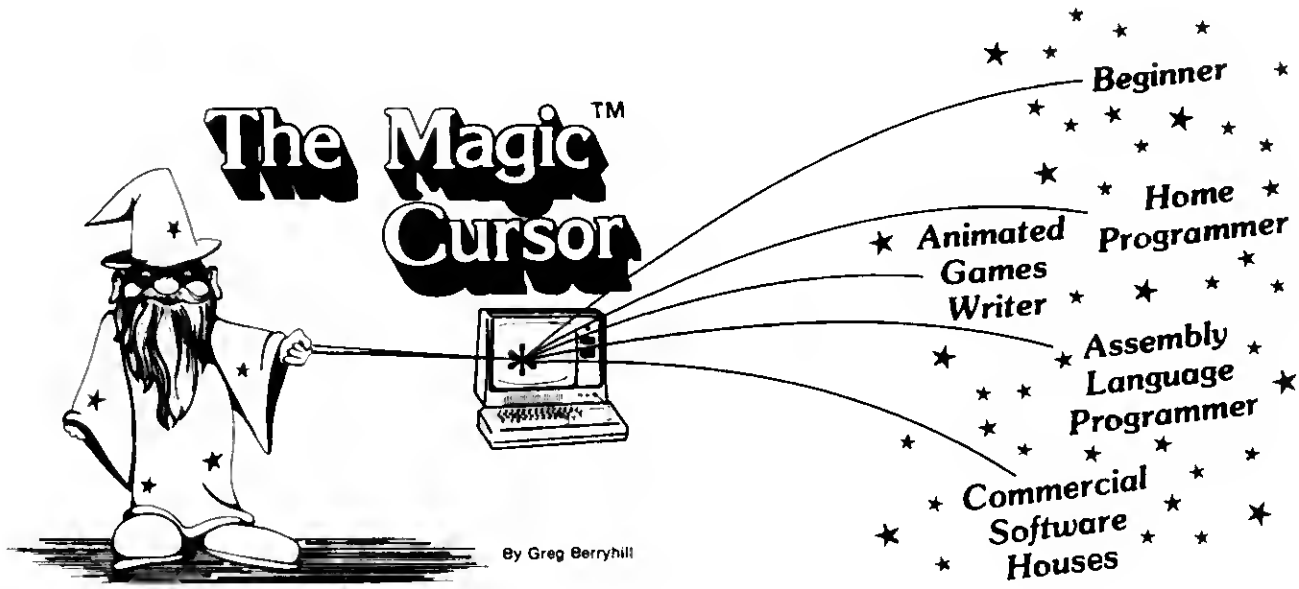
To this same feature another businessman replied, "That's un-American! Our country was founded on waste . . . having to buy more than you need. You are supposed to encourage them to move into a category where it will be to their advantage to buy more than they actually want."

Un-American? We don't think so. It is a fair pricing policy that gives us accuracy, reliability, speed and a lot of happy customers. ■

TRS-80* OWNERS:

*Trademark of Radio Shack, a Tandy Co.

- Let the computer write your "Basic" programs for you!
- Draw pictures, animated figures, data forms!
- Create a library of displays!
- Produce "Commercial" grade software!



The **Magic Cursor** is e Revolutionary Family of Products which provides a drametic new method of reproducing drewings end displays that you create on your screen. It makes both simple displays and complex interective date input forms. It stores a "Basic Program" on disk (or tape) ready for you to execute alone or as a subroutine.

It is availeble for any level 2, 16K or larger system with tape or disk. An optional version is now available with output files for assembly language programmers.

Be sure to pick out the system that fits your present needs and order it today, so you can take immediate advantege of this unique new product. You may upgrade your original copy by paying the difference and a moderate service charge.

THE BABY CURSOR allows you to easily create screens (including graphics) on your video. A powerful command then generates the BASIC instructions to recreate the screen. For the first time, a program for automatic generation of video display forms. (16K Tape or 16K Disk) **$24.95**

THE MAGIC CURSOR I additionally makes sophisticated Data Entry and Display easy. With Magic Cursor I you define the Data Entry or Display fields directly on your screen. The definition commands generate the BASIC instructions to implement the Data Entry and Display. The Magic Cursor I has commands which move, center and duplicate blocks of graphical or alpha/numeric displays. You can even justify text. (16K Tape Only) **$79.95**

THE MAGIC CURSOR II adds the power to write enimated gemes easily in BASIC. The Magic Cursor II allows you to reload previous screens either from memory or from Disk. You can then modify them and store either the modified screen or only the changes. (32K Disk only) **$99.95**

THE MAGIC CURSOR III will be available soon for the new Model II Computer (32K One or more Disks) **$149.95**

THE MAGIC CURSOR IV provides the features of Magic Cursor II but has output files for the essembly language programmer. (32K Disk only) **$99.95**

WRITE FOR OUR COMPLETE SOFTWARE CATALOG!

ORDER FROM

CUSTOM COMPUTER CENTER, INC. P.O. BOX 58042 / HOUSTON, TEXAS 77058 PHONE ORDERS: (800) 621-5199 / Illinois (800) 972-5855

Use our toll free number or order by mail. Master Charge, Visa, Certified Check or Money Order accepted. Personal checks require 14 days to clear. C.O.D. or collect calls not accepted. Software Guaranteed for replacement only. Prices subject to change without notice. Programs supplied on cassette tape. For disk versions, the cassette supplied will automatically create a disk file

Send Check or Money Order payable to: **CUSTOM COMPUTER CENTER, INC.** P.O. Box 58042 / Dept. B-C / Houston, Texas 77058

Attn: Jim Martens

QUAN.	DESCRIPTION	PRICE
	SYSTEM (Size/Tape or Disk)	
	TAX & HANDLING	
	TOTAL	

Handling Charge: $1.50 / Texas Res. add 6% Sales Tax.

MASTER CHARGE / VISA CARD # _____ EXPIRES _____

Name _____ Address _____ City _____ State _____ Zip _____

Bigger Is Better.

Expandable storage. Greater Versatility.

The Vista Model II* Disk Expansion system provides one, two or three drives, and adds up to one-half million bytes of storage.

You say you want more disk storage, more programming versatility . . . at a reasonable price. Say no more. Our Vista Model II Disk Expansion System may be just the answer for you. Choose from 1, 2 or 3 drives, already mounted. Each additional disk drive will add about one-half million bytes of on-line storage to your system. Buy only what you need now. As your needs grow, you can continue to expand your capacity by adding another drive. It's that simple.

But the best part we've saved for last. The price. It's cheaper than Radio Shack, yet our Model II will do everything Radio Shack's expansion system can do. The only difference is our Model II will keep on working long after most others have stopped. That's why we are justifiably proud of our product's high reliability. And our **120-day warranty**. Actually, it's not hard to stand behind a product — that works — if you know what we mean.

*Model II is a registered trademark of Radio Shack, a Tandy Company

COMPARE OUR INTRODUCTORY LOW PRICES

1-drive Expansion System	$1,000	3-drive Expansion System	$2,100
2-drive Expansion System	$1,550	Additional Drives Only	$ 525

NO WAITING • IMMEDIATE DELIVERY.

NAME _____

ADDRESS _____

CITY _____ STATE ___ ZIP _____

TELEPHONE ( ) _____

OPTIONAL

COMPANY AFFILIATION _____

TITLE _____

INTENDED APPLICATION _____

_____

PLEASE ☑ ONE:

☐ CHECK OR MONEY ORDER ☐ CREDIT CARD ☐ C.O.D.

PLEASE CHARGE MY CREDIT CARD: _____

☐ MASTER CHARGE ☐ BANK AMERICARD VISA

CARD NO. _____

EXPIRATION DATE _____

PRINT EXACT NAME OF CARD HOLDER _____

SIGNATURE _____

Item	Quantity	Price

Subtotal _____

Calif. Res. add 6% sales tax _____

Per-Unit Shipping and Handling _____

($5.00 UPS - $7.50 UPS Blue) _____

Total _____

Vista

CALL TOLL-FREE 800-854-8017

The Vista Computer Company 1401 Borchard Street • Santa Ana, California 92705 • 714/953-0523

More for Less.

Speed. Capacity. Price.

The Vista V-80 mini disk system is 8 times faster than the TRS-80, 23% more storage capacity, and costs only $395.

Compare our performance to Radio Shack's TRS-80*. Then match our price with theirs. Then decide which one is for you.

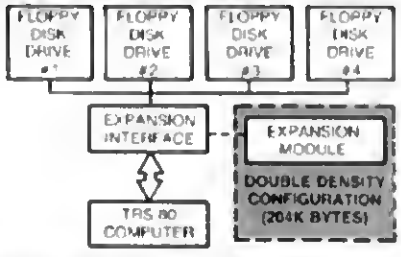
Features

- Vista offers 102K bytes to Radio Shack's 89K. That's 13K more bytes per drive for Vista.
- The V-80 operates at 12ms versus 40ms for TRS-80. Our drive can operate at 5ms, but only 50% of TRS-80 will operate at that speed; therefore, Vista has purposely set the access time at 12ms.
- Totally compatible with all available disk operating systems.
- Upgraded system. Increased storage and speed patch supplied at no charge by Vista.
- Drives are interchangeable for any location from Drive 0—thru Drive 3.
- Immediate Delivery.
- **120 Day Warranty**

Prices:
Single Drive System $395
Two Drive System $770
Four Drive System $1450

*TRS-80 is a registered trademark of Radio Shack, a Tandy Company

TYPICAL CONFIGURATION



Single Density Configuration (102K Bytes)

Vista Expansion Module

The expansion Module provides a double density modification to your current Radio Shack interface that allows you to format diskettes in either single or double density. In double density format, your Vista Drive increases your storage capacity up to 204K bytes on a single 40-track drive.

To insure the highest performance possible, without compromise, we recommend that you use Vista disk drives in conjunction with our Expansion Module.* For a demonstration on your system call **TOLL-FREE 800-854-8017**

Price $239.00 (includes all hardware and software)

**Vista cannot guarantee Radio Shack drives to operate 100% in double density.*

Vista

The Vista Computer Company 1401 Borchard Street • Santa Ana, California 92705 ■ 714/953-0523

This program uses sophisticated financial techniques to analyze possible investments.

Investment Analysis

Leslie E. Sparks
1014 Evergreen Dr.
Durhem, N.C. 27712

A microcomputer can be a powerful tool for analyzing capital investments. The following program shows you how to meke your computer a tool that is as sophisticated and reliable for economic analysis as those of any large companies.

Of course the computer cannot make your decisions for you, but it can provide you with important information in just minutes that will improve your decisions.

The basic criterion for this analysis program is Return on Investment (ROI). Although ROI is not the only factor to be considered in evaluating investments, ROI should strongly influence the final decision—especially if ROI is adjusted to include the value of money over time and inflation. The Program Listing does both.

Return on investment is the ratio of your net average annual cash flow to your total capital investment. Cash flow is your after tax profit plus depreciation. (Some authors define ROI as the ratio of after tax profit to total capital investment.)

ROI, expressed as cash flow divided by capital investment, is often called simple or engineer-

Program Listing

```
10 REM INVESTEMENT ANALYSIS BY L.E. SPARKS
20 REM WRITTEN IN TRS80 LEVEL II BASIC 27 AUG 1979
30 REM WITH 16K RAM YOU CAN LOOK AT A PROJECT LIFE OF ABOUT
40 REM 15 YEARS A LOAN LIFE OF 15 YEARS AND 12 PAYMENT/YEAR
50 REM INPUT VARIABLES--
60 REM D=DEPRECIABLE BASE $, N= PROJECT LIFE
70 REM SV =SALVAGE VALUE
80 REM L1=AMOUNT OF LOAN,I=INTEREST RATE %
90 REM NL=LIFE OF LOAN YEARS, NP=NUMBER OF PAYMENTS/YEAR
100 REM R=YEARLY REVENUE $, O=YEARLY OPERATING COST $
110 REM IR=ESCLATION RATE FOR REVENUE %
120 REM IO=ESCLATION RATE PER YEAR FOR OPERATING COSTS %
130 REM  TC=TAX CREDIT FOR INVESTMENT %(10% IS DEFAULT)
140 REM T=MARGINAL TAX RATE % (50% IS DEFAULT)
150 REM G =ANNUAL GNP DEFLATOR (8% ASSUMED)
160 DIM A$(3):REM A$ IS A STRING FOR LABELING DEPRECIATION METHOD
170 A$(1)="STRIGHT LINE DEPRECIATION"
180 A$(2)="SUM OF YEARS DIGITS DEPRECIATION"
190 A$(3)="DECLINING BALANCE DEPRECIATION"
200 REM JT IS A FLAG WHICH IS =1 IF INVESTMENT TAX CREDIT IS NOT
210 REM TAKEN IN 7 YEARS.  A WARNING IS PRINTED IF JT=1
220 REM TL IS THE TOLERANCE FOR TRIAL AND ERROR SOLUTION SET AT 0.001
230 TL=0.001
240 REM-----------------------------------------
250 DEFINT J  REM DEFINE ALL J VARIABLES AS INTEGERS
260 CLS
270 JD=0
280 JF=0
290 JT=0
300 REM JD AND JF ARE FLAGES USED TO DETERMINE DEPRECIATION METHOD
310 PRINT"INVESTMENT ANALYSIS FOR TRS 80"
320 PRINT"ENTER INFORMATION ASKED FOR "
330 PRINT"PRESS <ENTER> AFTER EACH ENTRY"
340 PRINT"CURRENT VALUE IS SHOWN IN ()
350 PRINT"IF YOU WANT TO USE VALUE IN () JUST PRESS <ENTER>"
360 PRINT"IF YOU MAKE AN ERROR  CONTINUE ENTERING DATA"
370 PRINT"YOU WILL BE GIVEN THE OPPORTUNITY TO CORRECT DATA"
380 PRINT"BEFORE CALCULATIONS BEGIN"
390 PRINT
400 REM SET UP DEFUALTS
410 G=8
420 TC=10
430 T=50
440 PRINT"ENTER YEARLY REVENUE $ (";R;")";
450 INPUT R
460 PRINT "ENTER YEARLY OPERATING COSTS $ (";O;")";
470 INPUT O
480 PRINT "ENTER ESCALTION RATE FOR REVENUE % (";IR;")",
490 INPUT IR
500 PRINT"ENTER ESCALATION RATE FOR COSTS % (";IO,")";
510 INPUT IO
520 PRINT"ENTER DEPRECIABLE BASE $ (";D,")";
530 INPUT D
540 PRINT"ENTER SALVAGE VALUE (";SV;")",
550 INPUT SV
560 IF SV>D PRINT"SALVAGE VALUE > DEPRECIABLE BASE" ELSE GOTO590
570 INPUT"IS THIS CORRECT ";Y$
580 IF LEFT$(Y$,1)="N" GOTO 520
590 PRINT"ENTER PROJECT LIFE YEARS (";N;")";
600 INPUT N
610 PRINT"ENTER AMOUNT OF LOAN $ (";L1;")",
620 INPUT L1
630 IF L<=D THEN 660  ELSE PRINT"LOAN > DEPRECIABLE BASE"
640 INPUT"IS THIS CORRECT ";Y$
650 IF LEFT$(Y$,1)="Y" THEN 660  ELSE GOTO 620
660 PRINT"ENTER ANNUAL INTEREST RATE % (",I,")";
670 INPUT I
680 REM CHECK TO SEE IF I IS %
690 IF I>.999999 THEN 720
700 PRINT"INTEREST RATE SHOULD BE %"
710 GOTO 660
720 PRINT "ENTER LIFE OF LOAN IN YEARS (";NL;")";
730 INPUT NL
740 IF NL<=N THEN GOTO790
750 PRINT"THE PROJECT LIFE IS LESS THAN THE LIFE OF LOAN"
760 INPUT" IS THIS CORRECT ";Y$
```

At Last! Affordable
WORD PROCESSING
Made Possible by TRS-80™

NEW SCRIPSIT™ Word Processing Software for 16K Level II TRS-80s.
$69.95 Cassette*
$99.95 Disk*

EDIT, DELETE, MODIFY— then print it error-free, 45 characters per second!

NEW UPPER AND LOWER CASE KIT. Available for new or existing systems.
$99 Installed*

NEW TRS-80 DAISY WHEEL PRINTER. Clean and readable printing, like the very finest electric typewriters.
$2999*

RADIO SHACK BRINGS YOU A COMPLETE WORD PROCESSING SYSTEM FOR TRS-80 MODEL I COMPUTERS.

Radio Shack smashes another computer "cost barrier" with the new TRS-80 Word Processing System. The system includes our new SCRIPSIT software, Upper/Lower Case Kit and Daisy Wheel Printer. Add it to any 16K Level II TRS-80, or buy a complete system. Once you've tried it, you may never want to use a typewriter again!

The new SCRIPSIT software lets you compose letters and documents of all types on TRS-80's screen in upper case, or upper and lower case with the new Upper/Lower Case Kit. You can move words or entire paragraphs, insert, delete and edit to your heart's content! SCRIPSIT gives you automatic page numbers, page headings and footnotes and makes it easy to indent paragraphs, change line widths, and center your text horizontally or vertically. Advanced features include justification, hyphenation, global search/replace, and variable screen width. On-going reports, form letters and text with print commands can be stored on TRS-80 cassettes or diskettes for use or revision at any time.

SCRIPSIT software includes an audio cassette course that makes anyone a proficient word processing operator.

PRINT ALL OF THE "ORIGINALS" YOU NEED, FAST AND ERROR-FREE!

Our new WP-50 Daisy Wheel Printer is fast and gives you the same quality of the finest electric typewriters — carbon film ribbon and all! Or, if your job doesn't require "letter" quality, a TRS-80 system with a dot matrix, u/lc printer costs even less.

A complete TRS-80 cassette system with Word Processing Software, Upper/Lower Case Kit and a dot matrix printer is yours for just $2,046.95*. Or choose a really deluxe system with the WP-50 Printer and two floppy disks that store eight hours of 50 WPM typing for only $5,492.95*.

Sound exciting? You bet it is! Visit your nearest Radio Shack outlet or write for details.

*Retail prices may vary at individual stores and dealers.

Mail to: Radio Shack, Dept. CMA-450
1300 One Tandy Center
Fort Worth, Texas 76102

I'd Like to Know More!

☐ Send details on TRS-80 Word Processing and the 24-page TRS-80 Catalog #RSC-3.

☐ Have a representative contact me.

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

I Own/Use a TRS-80 ☐ Yes ☐ No

Model _____

Radio Shack®

The biggest name in little computers™

A DIVISION OF TANDY CORPORATION • FORT WORTH, TEXAS 76102

✓5

```
Subroutine Loan Payment
Age = 0
Sum = 0
Amt = Loan
Npay = life * Npayyr
Inrate = Inyr/Npayyr
Val = (Inrate + 1.0)↑ (Npay)
Pay = (Inrate * Val *Loan)/(Val - 1.0)
Do while Age<Life
     Inpaid = Inrate * Amt
     Pripaid = Pay - Inpaid
     Amt = Amt - Pripaid
     Sum = Sum + Pripaid
     Age = Age + 1
End do
if Sum - Amt ≠ Then Last Pay = Pay + Sum Endif
End subprogram
```

Figure 1. Algorithm for calculating loan payment.

ing ROI. Its virtue is that it is simple to calculate. Its major deficiency is that simple ROI does not consider the timing of the cash flow. Thus, the simple ROI is the same for an investment that yields a cash flow of $100 a year for ten years and a second investment with zero cash flow for nine years and $1000 for the tenth year.

I'm sure you'll egree that the first of these two cash flows is better than the second, because of the value of money over time.

A dollar today can be invested and earn interest. Thus, a dollar today will be worth $1.10 next year if we invest it at 10 percent interest. For this reason, a dollar today is worth more than a dollar next year. (Time value of money should not be confused with inflation. Even with zero inflation, a dollar today is worth more than a dollar next year because we could invest today's dollar and earn interest.)

To account for the time value of money, we must reduce, i.e. discount, the value of future year dollars by the amount that this year's dollars could earn. For example, if we can earn 10 percent interest today, that dollar will be worth $1.10 in one year. Thus, it takes a cash flow of $1.10 next year to equal a cash flow of $1.00 this year.

The investment figure that reflects the time value of money is called Discount Cash Flow Return On Investment (DSCF ROI). It can also be called interest rate of return, profitability index and investors' method. Essentially, DSCF ROI is the interest rate that reduces all future cash flow so that their sum equals your capital investment.

The DSCF ROI can change the rank of your possible invest-

```
770 IF LEFT$(Y$,1)="Y" THEN 790
780 GOTO 590
790 PRINT"ENTER NUMBER OF LOAN PAYMENTS PER YEAR (";NP;")";
800 INPUT NP
810 PRINT"ENTER INVESTMENT TAX CREDIT RATE (10% ASSUMED)(";TC;")";
820 INPUT TC
830 IF TC>=1 THEN 860
840 PRINT"INVESTMENT TAX CREDIT RATE SHOULD BE IN %"
850 GOTO810
860 PRINT"ENTER INCOME TAX RATE % (50% ASSUMED) (";T;")";
870 INPUT T
880 IF T>=1 THEN  910
890 PRINT"INCOME TAX RATE SHOULD BE IN %"
900 GOTO 860
910 PRINT"ENTER GNP DEFLATOR % (8% ASSUMED)";G;")";
920 INPUT G
930 IF G>=1 THEN 960
940 PRINT "GNP DEFLATOR SHOULD BE IN %"
950 GOTO 910
960 REM-----------------------------------------------
970 REM END OF INPUT NOW PRINT IT AND SEE IF OK
980 CLS
990 PRINT"THE INPUT DATA ARE AS FOLLOWS"
1000 PRINT"YEARLY REVENUE                $";R
1010 PRINT"YEARLY OPERATING COSTS        $";O
1020 PRINT"ESCALATION RATE FOR REVENUE    ";IR;"%"
1030 PRINT"ESCALATION RATE FOR COSTS      ";IO;"%"
1040 PRINT"DEPRECIABLE BASE              $";D
1050 PRINT"PROJECT LIFE                   ";N;"YEARS"
1060 PRINT"AMOUNT OF LOAN                $";L1
1070 PRINT"ANNUAL INTEREST RATE           ";I;"%"
1080 PRINT"LIFE OF LOAN                   ";NL;"YEARS"
1090 PRINT"NUMBER OF PAYMENTS/YEAR        ";NP
1100 PRINT"TAX CREDIT RATE                ";TC;"%"
1110 PRINT"INCOME TAX RATE                ";T;"%"
1120 PRINT"GNP DEFLATOR                   ";G;"%"
1130 INPUT"ARE THESE CORRECT YES OR NO ";Y$
1140 IF LEFT$(Y$,1)<>"N" THEN 1170
1150 PRINT"ENTER INCORRECT DATA PRESS <ENTER> IF VALUE IN () IS OK"
1160 GOTO 240
1170 REM ASK FOR METHOD OF DEPRECIATION
1180 CLS
1190 JD=0
1200 JF=0
1210 JN=3
1220 PRINT"YOU MAY SELECT ONE OF THE FOLLOWING  METHODS OF DEPRECIATION OR"
1230 PRINT"YOU CAN HAVE THE COMPUTER CALCULATE THE ANALYSIS FOR ALL THREE"
1240 PRINT"OF THE FOLLOWING METHODS OF DEPRECIATION"
1250 PRINT"1. STRAIGHT LINE 2. SUM OF YEARS DIGITS"
1260 PRINT"3. DECLINING BALANCE "
1270 PRINT"TO SELECT A METHOD OF CALCULATING DEPRECIATION"
```

```
Subroutine Declining Balance Depreciation
Age = 0
Rate = 2/Life
Sum = 0
BookValue = Cost
AnDep = 0
AcumDep = 0
Do while Age<Life
    Age = Age + 1
    AnDep = BookValue * Rate
    If BookValue - AnDep<Salvage
      then AcumDep = AcumDep + AnDep
      BookValue = BookValue - AnDep
      Else AnDep = BookValue - Salvage
      BookValue = Salvage
      If Age ≠ Life then exit End if
    End if
End do
End Subroutine
```

```
Subroutine Straight Line Depreciation
Age = 0
Value = Cost - Salvage
AcumDep = 0
Do while Age<Life
    AnDep = Value/Life
    AcumDep = AcumDep + AnDep
    Age = Age + 1
End do
End subroutine

Subroutine Sum of Years Digits
Age = 0
Value = Cost - Salvage
AnDep = 0
AcumDep = 0
NO = Life*(Life + 1)/2
Do while Age<Life
    AnDep = ((Life - Age)/NO)*Val
    AcumDep = AcumDep + AnDep
    Age = Age + 1
End do
End subroutine
```

Figure 2. Algorithms for depreciation

```
Subroutine Cash Flow
Year = 0
Taxcredit = 0.1*Depbase
Taxloss = 0
Do while Year<Life
    Revenue = Revenue*(1 + Revescalation)
    Cost = Cost *(1 + Costescalation)
    Grossprofit = Revenue - Cost - Interest - Dep
    If Grossprofit >0 then
      Grossprofit = Grossprofit - Taxloss
      If Grossprofit>0 then
        Tax = TaxRate*Grossprofit
        Tax = Tax - TaxCredit
        If Tax>0 then TaxCredit = 0
          Else Tax = 0
          TaxCredit = - Tax
        End If
        Else Tax = 0
        Taxloss = - Grossprofit
      End If
      Else Tax = 0
      Taxloss = Taxloss - Grossprofit
    End if
    Cashflow = Grossprofit + Dep
    Year = Year + 1
End do
End Subroutine
```

Figure 3. Cash Flow Subroutine.

ments from the one obtained using simple ROI. However, the major disadvantage of DSCF ROI is that trial and error is the only way to calculate it.

Before going further, I would like to define some terms so that we're all talking the same language.

Inflation is the rise in the average level of *all* prices.

Escalation is the rise in the price of *a single* commodity or service. Some commodities may escalate without *inflation*. Escalation rates vary from commodity to commodity and from service to service.

Current year dollars are those received in a specific year. The current year dollars for any two years may or may not have the same purchasing power. Current year dollars are a "rubber" ruler that cannot be used as an absolute measure of cash flow in different years.

Constant dollars are *current year dollars* referenced according to their purchasing power to some base year. The base year used in the Program Listing is year zero of the investment. Constant dollars are the absolute ruler for measuring cash flow in different years. Constant dollars are calculated by discounting *current year dollars* by the inflation rate back to the base year. (Discounting *current year dollars* to account for inflation should not be confused with discounting future year cash flow to account for the time value of money.)

We all know that inflation constantly reduces the value of the dollar. Inflation also has a major impact on the profitability of investments. Yet, generally, inflation is not considered in investment analysis.

To evaluate a potential investment we should escalate the revenues and operating costs, calculate current year cash flow, discount the inflation rate of current year cash flow and then calculate the DSCF ROI from the constant dollar cash flow.

When this is done, we can compare the DSCF ROI for various investment possibilities.

The Program

Now that we have established the criteria, let's build an analysis program. Our program should calculate simple ROI, DSCF ROI in current year dollars and DSCF ROI in constant dollars. What our program must do:

1. Read in data.
2. Check data for correctness.
3. Calculate payment schedule for outstanding loans.
4. Calculate yearly depreciation.
5. Calculate revenue for each year, including escalation.
6. Calculate yearly operating costs including escalation.
7. Calculate yearly before tax profit.
8. Calculate income tax due.
9. Subtract tax credit if any.
10. Calculate after tax income.
11. Calculate current year dollar cash flow.
12. Calculate constant dollar cash flow.
13. Calculate simple ROI using current year dollars.

```
1280 PRINT"ENTER THE NUMBER OF THE METHOD YOU WANT
PRESS <ENTER> IF YOU WANT THE COMPUTER TO CALCULATE ALL 3"
1290 INPUT JD
1300 IF JD>3 THEN PRINT" PLEASE YOU ENTERED AN INCORRECT NUMBER":JD=0:GOTO 1230
1310 IF JD<>0 THEN JF=1 ELSE JD=1 · .
1320 REM FIRST DIMENSION ARRAYS
1330 REM CF(I)=CASH FLOW FOR ITH YEAR.D(I)=DEPRECIATION FOR ITH YEAR
1340 REM R(I)=REVENUE FOR ITH YEAR D(I)=OPERATING COST FOR ITH YR
1350 REM IP(I)=INTEREST PAID FOR ITH YEAR. TP(I)=TAX PAID FOR ITH YR
1360 REM TC(I)=TAXES PAID IN ITH YEAR.P(I)=BEFORE.TAX PROFIT
1370 REM PT(I)=AFTER TAX PROFITS FOR ITH YEAR
1380 REM RO IS THE SIMPLE ROI,R1 IS DISCOUNTED CASH FLOW ROI NOT CORRECTED FOR INFLATION
1390 REM R2 IS THE DSCF ROI CORRECTED FOR INFLATION
1400 IF JG=1 THEN 1430
1410 DIM CF(N+1,3),D(N+1,3),R(N+1),D(N+1),IP(N+1),TC(N+1,3),P(N+1,3),PT(N+1,3),IM(ML*NP+1),RO(3),R1(3),R2(3),JT(3)
1420 REM NOW AMORIZE LOAN
1430 CLS
1440 PRINT"PLEASE WAIT IT WILL TAKE 1 TO 2 MINUTES FOR CALCULATIONS"
1450 REM I1 IS INTEREST RATE PER MONTH I/NP
1460 I1=I/NP/100.000 :REM THIS IS MONTHLY INTEREST
1470 V=(I1+1.0000)^(NP*ML)
1480 P=L1*V*I1/(V-1 )
1490 REM NOW CALCULATE PAYMENT SCHEDULE
1500 S=0
1510 S1=0
1520 N3=1
1530 S2=0
1540 A=L1
1550 J1=1
```

✔ Reader Service—see page 147

```
1560 N2=NL*NP
1570 IP(J1)=0
1580 J2=1
1590 I2= INT(I1*M*100)/100 : REM INTEREST TO NEAREST CENT
1600 IM(N3)=I2
1610 PP=P-I2
1620 IP(J1)=IP(J1)+I2
1630 S=S+PP
1640 R=R-PP
1650 IF N3=N2 GOTO 1730
1660 IF J2=12 THEN GOTO 1700
1670 N3=N3+1
1680 J2=J2+1
1690 GOTO1590
1700 J1=J1+1
1710 N3=N3+1
1720 GOTO 1570
1730 REM FINISHED
1740 REM NOW DEPRECIATE THE INVESTMENT
1750 REM PROGRAM WILL EITHER CALCULATE DEPRECIATION ON
1760 REM USER SPECIFIED METHOD OR FOR
1770 REM STRAIGHT LINE, SUM OF YEARS DIGITS, DECLINING BALANCE
1780 REM AND DOUBLE DECLINING BALANCE
1790 REM JD IS FLAG FOR METHOD OF DEPRECIATION
1800 REM JD=1 STRAIGHT LINE, JD=2 SUM OF YEARS DIGITS
1810 REM JD=3 DECLINING BALANCE
1820 D1=D-SV
1830 REM CHECK TO SEE IF USER HAS SPECIFIED METHOD
1840 IF JF=0 THEN JD=1 ELSE JN=1
1850 ON JD GOTO  1860 ,1940 ,2030
1860 REM DEPRECIATION FOR STRAIGHT LINE JD=1
1870 FOR J6=1 TO N
1880    D(J6, JD)=D1/N
1890    SD=SD+D(J6, JD)
1900 NEXT J6
1910 IF JF<>0 THEN 2170
1920 JD=JD+1
1930 GOTO1850
1940 REM SUM OF YEARS DIGITS DEEPRECIATION
1950 ND=N*(N+1)/2
1960 FOR J6=1TON
1970    D(J6, JD)=((N-(J6-1))/ND)*D1
1980    ZZ=ZZ+D(J6, JD)
1990 NEXT J6
2000 IF JF<>0 THEN 2170
2010 JD=JD+1
2020 GOTO 1850
2030 REM NOW CALCULATE DECLINING BALANCE
2040 R1=2/N
2050 BV=D
2060 SD=0 REM SD IS THE ACCUMULATED DEPRECIATION
2070 FOR J6=1TON
2080    D(J6, JD)=BV*R1
2090    IF BV-D(J6, JD)<SV THEN 2130 ELSE D(J6, JD)=BV-SV
2100    BV=SV
2110    SD=SD+D(J6, JD)
2120    GOTO 2160
2130    BV=BV-D(J6, JD)
2140    IF BV<= SV THEN BV=SV
2150    SD=SD+D(J6, JD)
2160 NEXTJ6
2170 REM DEPRECIATION CALCULATIONS ARE COMPLETE
2180 IF JF=0 THEN JD=1 ELSE JN=1
2190 REM NOW CALCULATE TAXES PROFITS AND CASH FLOW
2200 IR=IR/100
2210 TC=TC/100
2220 T=T/100
2230 I0=I0/100
2240 FORJ1=JD TO JN
2250    TL=0
2260    TX=TC*D
2270   FOR J2=1TON
2280         R(J2)=R*(1+IR)^J2
2290          D(J2)=D*(1+IR)^J2
2300          P(J2, J1)=R(J2)-D(J2)-IP(J2)-D(J2, J1)
2310 REM ADD IN REMAINING SALVAGE VALUE IF DECLINING BALANCE
2320 REM DEPRECIATION USED AND IF DEPRECIATED BOOK VALUE IS
2330 REM GREATER THAN SALVAGE VALUE AND IF THIS IS LAST YR
2340          IF J2=N THEN P(J2,3)=P(J2,3)+BV-SV
```

```
2350          IF P(J2,J1)<=0 TC(J2,J1)=0
2360          IF P(J2,J1)<=0 TL=TL-P(J2,J1) GOTO2510
2370          G2=P(J2,J1)-TL
2380          IF G2<0 TL=-G2 TC(J2,J1)=0 GOTO2510
2390          P(J2,J1)=G2
2400 REM PRETAX PROFIT =PRETAX PROFIT - TAXLOSS CARRY FORWARD
2410          TL=0
2420          TC(J2,J1)=T+G2
2430 REM TAKE INVESTMENT TAX CREDIT
2440 REM CHECK TO SEE IF INVESTMENT TAX CREDIT TAKEN BEFORE
2450 REM NUMBER OF YEARS EXCEEDS 7
2460          IF J2>7 AND TX>0 THEN JT(J1)=1
2470          TC(J2,J1)=TC(J2,J1)-TX
2480          IF TC(J2,J1)>0 TX=0
2490          IF TC(J2,J1)<0 TX=-TC(J2,J1)
2500          IF TC(J2,J1)<0 TC(J2,J1)=0
2510          PT(J2,J1)=P(J2,J1)-TC(J2,J1)
2520          CF(J2,J1)=PT(J2,J1)+D(J2,J1)
2530    NEXTJ2
2540 NEXTJ1
2550 TL=.001
2560 REM NOW CALCULATE SIMPLE ROI
2570 FOR J2=JDTOJN
2580    PO(J2)=0
2590    FOR J1=1TON
2600          PO(J2)=PO(J2)+CF(J1,J2)
2610    NEXT J1
2620    RO(J2)=PO(J2)/(N*D)
2630 NEXT J2
2640 REM NOW CLACULATE DCCF ROI THIS IS TRIAL AND ERROR CALCULATION
2650 FOR J2=JDTOJN
2660   L=-1
2670   H= 2
2680    S=0
2690    R1(J2)=(H+L)/2
2700    FOR J1=1TON
2710          S=S+CF(J1,J2)*(1+R1(J2))(-J1 REM DISCOUNT FACTOR
2720    NEXT J1
2730    ER=(S-D)/D
2740    IF ABS(ER)<=TL THEN 2780
2750   IF ABS(H-L)<.0001 THEN PRINT"FAILED TO CONVERGE " GOTO2780
2760    IF ER>0THEN L=R1(J2) ELSE H=R1(J2)
2770    GOTO 2680
2780 NEXT J2
2790 REM NOW CALCULATE DCCF ROI ACCOUNTING FOR INFLATION
2800 G=G/100
2810 FOR J2=JDTOJN
2820   L=-1
2830   H=2
2840    S=0
2850    R2(J2)=(H+L)/2
2860    FOR J1=1TON
2870          DC=CF(J1,J2)*(1+G)(-J1:REM DISCOUNT CASH FLOW FOR INFLATION
2880          S=S+DC*(1+R2(J2))(-J1
2890    NEXTJ1
2900    ER=(S-D)/D
2910    IF ABS(ER)<=TL THEN 2950
2920    IF ER>0 THEN L=R2(J2) ELSE H=R2(J2)
2930   IF ABS(H-L)<.001 THEN PRINT" FAILED TO CONVERGE " GOTO2950
2940    GOTO 2840
2950 NEXTJ2
2960 REM NOW PRINT IT OUT
2970 REM FIRST PRINT OUT VARIOUS ROI
2980 CLS
2990 FORJ1=JDTOJN
3000   PRINT"SIMPLE ROI FOR ";A$(J1);"  ";
3010   PRINT USING "##.##%";RO(J1)*100
3020   IF JT(J1)=1 THEN PRINT"WARNING   TAX CREDIT NOT TAKEN IN 7 YEARS"
3030 NEXT J1
3040 FORJ1=JDTOJN
3050   PRINT"DSCF ROI FOR "A$(J1)" ";
3060   PRINT USING "##.##%";R1(J1)*100
3070 NEXT J1
3080 FOR J1=JDTOJN
3090   PRINT USING"DSCF ROI DISCOUNTED FOR ##.##% INFLATION ";G*100
3100   PRINT"FOR "A$(J1) ;
3110   PRINT USING" ##.##%";R2(J1)*100
3120 NEXTJ1
3130 INPUT"PRESS <ENTER> TO SEE DETAILS ";ZZ
```


Reader Service—see page 147

```
3140 CLS
3150 REM NOW PRINT IT OUT ON CRT
3160 CLS
3170 FORJ1=1TON
3180   FORJ2=JDTOJN
3190          CLS
3200          PRINT"DETAILS FOR YEAR # ";J1;" FOR ";A$(J2)
3210           PRINT"REVENUE";TAB(45);"$";R(J1)
3220           PRINT"OPERATING COST";TAB(45),"$";O(J1)
3230           PRINT"INTEREST PAID";TAB(45),"$";IP(J1)
3240           PRINT"DEPRECIATION FOR ";TAB(45);"$";D(J1,J2)
3250           PRINT"PRETAX INCOME";TAB(45);"$";P(J1,J2)
3260          PRINT"TAX PAID",TAB(45);"$",TC(J1,J2)
3270           PRINT"AFTERTAX INCOME",TAB(45);"$",PT(J1,J2)
3280           PRINT"CURRENT YEAR  DOLLARS CASH FLOW ";TAB(45);"$",CF(J1,J2)
3290           PRINT"DISCOUNTED CASH FLOW ";TAB(45);"$";CF(J1,J2)*(1+R1(J2))[-J1
3300           CC=D(J1,J2)+(PT(J1,J2)*(1+G)[-J1)
3310           PRINT"CASH FLOW IN CONSTANT $ WITH ";G*100;"% INFLATION";TAB(45);"$",CC
3320           PRINT"DISCOUNTED CONSTANT DOLLAR CASH FLOW";TAB(45);"$";CC*(1+R2(J2))[-J1
3330   INPUT"PRESS <ENTER> TO SEE MORE ";DD
3340    NEXT J2
3350 NEXT J1
3360 INPUT"DO YOU WANT HARD COPY ";Y$
3370 IF LEFT$(Y$,1)="Y" THEN 3400 ELSE INPUT"DO YOU WANT TO CHANGE INPUT";Y$
3380 IF LEFT$(Y$,1)="N" THEN STOP ELSE JG=1
3390 GOTO 230
3400 REM HARD COPY OUTPUT OF EVERYTHING
3410 POKE 16424,48
3420 POKE 16425,0
3430 LPRINT TAB(20);"INPUT DATA "
3440 LPRINT
3450 LPRINT"YEARLY REVENUE ";TAB(45);"$";R
3460 LPRINT"YEARLY OPERATING COST ";TAB(45);"$";O
3470 LPRINT"ESCALATION FOR REVENUE";TAB(45);IR*100;"%"
3480 LPRINT"ESCALATION FOR COSTS";TAB(45);IO*100;"%"
3490 LPRINT"DEPRECIABLE BASE ";TAB(45),"$";D
3500 LPRINT"SALVAGE VALUE ";TAB(45),"$";SV
3510 LPRINT"AMOUNT OF LOAN ";TAB(45);"$",L1
3520 LPRINT"ANNUAL INTEREST RATE ";TAB(45);I;"%"
3530 LPRINT"LIFE OF LOAN";TAB(45);NL;"YRS"
3540 LPRINT"NUMBER OF PAYMENTS/YR";TAB(45);NP
3550 LPRINT"PROJECT LIFE ";TAB(45);N
3560 LPRINT"INVESTMENT TAX CREDIT RATE ";TAB(45);TC*100;"%"
3570 LPRINT"INCOME TAX RATE ";TAB(45);T*100;"%"
3580 LPRINT"GNP DEFLATOR ";TAB(45);G*100,"%"
3590 LPRINT
3600 FOR J1= JD TO JN
3610    LPRINT" SIMPLE ROI FOR ";A$(J1);TAB(45),RO(J1)*100;"%"
3620    LPRINT" DCCF ROI FOR ";A$(J1);TAB(45);R1(J1)*100;"%"
3630    LPRINT" DCCF ROI CORRECTED FOR INFLATION FOR " ;A$(J1);TAB(45),R2(J1)*100;"%"
3640 NEXT J1
3650 LPRINT CHR$(11)
3660 INPUT"PRESS ENTER FOR NEXT PAGE";XX
3670 REM HARD COPY FOR DETAILS
3680 FOR J1= 1 TO N
3690       LPRINT "DETAILS FOR YEAR # ";J1
3700       LPRINT "REVENUE ";TAB(47);R(J1)
3710       LPRINT "OPERATING COST",TAB(47);O(J1)
3720       LPRINT"INTEREST PAID";TAB(47);IP(J1)
3730    FOR J2=1TOJN
3740           LPRINT"ANALYSIS FOR ";A$(J2)
3750           LPRINT"DEPRECIATION ";TAB(47);"$";D(J1,J2)
3760           LPRINT"PRETAX PROFIT";TAB(47);"$";P(J1,J2)
3770           LPRINT"TAX PAID";TAB(47);"$";"$";TC(J1,J2)
3780           LPRINT"AFTER TAX INCOME";TAB(47);"$";PT(J1,J2)
3790           LPRINT"CURRENT YEAR CASH FLOW ";TAB(47),"$";CF(J1,J2)
3800           LPRINT"DISCOUNTED CURRENT DOLLAR CASH FLOW";TAB(47);"$";CF(J1,J2)*(1+R1(J2))[-J1
3810           CC=CF(J1,J2)*(1+G)[-J1
3820           LPRINT"CONSTANT DOLLAR CASH FLOW FOR ";G*100;"% INFLATION";TAB(47);"$",CC
3830           LPRINT"DISCOUNTED CONSTANT DOLLAR CASH FLOW",TAB(47);"$";CC*(1+R2(J2))[-J1
3840    NEXTJ2
3850    LPRINT CHR$(11)
3860    INPUT"PRESS <ENTER> FOR NEXT PAGE";ZZ
3870 REM IF YOU DO NOT WANT PAGING REMOVE THE ABOVE TWO STATEMETNS
3880 NEXTJ1
3890 INPUT"DO YOU WANT TO CHANGE INPUT ";Y$
3900 IF LEFT$(Y$,1)="N" THEN STOP ELSE JG=1
3910 GOTO 230
```

TBS-80 GENERAL ACCOUNTING SYSTEMS. ONE STEP BEYOND.

Your TRS-80™ microcomputer is not a toy. These **TBS-80 general accounting systems** aptly demonstrate the power of your computer. **ANALYSIS PAD** by Del Jones is the epitome of first-class programming in business applications. Requiring 48K, and one disk with a printer recommended, this columnar calculator gives the user tremendous flexibility in data entry, enabling the user to create 30 or more columns and rows. Enter your own column and row labels. Enter your data by row or column or directly onto screen display via edit mode. Move, swap, delete, and add rows or columns. Create new pads by stripping relevant data from old files. You never have to key in data twice. But, more important than the powerful data manipulation provided, add, subtract, multiply and divide one column by another and put results in another column. Perform up to six calculations on one column and even define one column to be a constant. The calculation routine you create can be saved and reused. Print out the entire pad in four column segments to fine or serial printer. **ANALYSIS PAD** was originally advertised for 32K tape at $32.50. Since then, it has been totally rewritten and expanded to its present 48K disk only form and sells for $49.50. It is easily worth twice as much. You have to see it to believe it.

CHECK REGISTER ACCOUNTING SYSTEM, adapted for the TRS-80 by Dale Kubler and originally written by O.E. Dial, is the most comprehensive check-balancing program written. Requiring 32K, two disks and printer, this program does much more than just balance and reconcile your checkbook. It enables you to define up to 60 account names and will generate monthly summaries of all accounts with monthly and year-to-date totals. Single-entry input allows the user to disperse one transaction over several accounts and to make a 64-character note on each transaction. Checks can be printed out after data has been entered. Aside from the Statement of Accounts, **CRAS** also generates the following reports: Check Register for any Month, Notes to Check Register, Income/Expense Distribution, Statement of Selected Accounts, Bank Reconcile Statement and Suspense File. The Suspense file is an extra feature where you can make notes to yourself for any month in the year. **CRAS** will make both you and your account happy and it sells for $49.50.

CHECKBOOK II by Alan Meyers is the finest program of its kind yet published. With superb graphic screen displays, it does everything necessary to keep your checkbook balanced. Data is input directly into a five-column screen display with a field for alpha or numeric codes. Editing is done easily for changes in any or all columns. **CHECKBOOK II** will accurately balance and reconcile your checkbook, handling balances up to $1,000,000. Your balance brought forward is always in memory. Outstanding checks are listed and easily saved. You can also search for an entry by any field except amount, and all checks with matching entries will be displayed and totaled. A numeric sort routine is included. Screen prints can be made to a line printer from almost any point in the program. In addition, the 32-48K version can write files to disk. This, and the 16K version, are included on the same tape. For $18.50, **CHECKBOOK II** is the bottom line in personal checkbook programs. A disk version of this program is available for $28.50.

BUDGET II (not yet released) by Alan Meyers, takes off where CHECKBOOK II ends. Written exclusively for either disk or tape based computers, this program enables the user to set up 20 account names with four character codes for each, that correspond to the codes used in Checkbook II. Each account can be tagged income or expense and whether it is fixed or not. Set your monthly budget and balance it. Disperse your cash account over the other accounts. Checkbook II data is brought in and summarized by account and compared to amount budgeted. Year-to-date totals are included in monthly summary. Year Summary gives monthly and year totals for each account at a glance. Forecast feature enables user to enter rate of inflation and income increase to see financial standing after 12 months. Review enables user to go back and look at months previously summarized. Flashy graphics and much more. For 16K and 32K tape, **BUDGET II** sells for $24.50. For 32K up disk, $34.50. If you have CHECKBOOK II, you will want this program.

TBS has other incredible software for Tandy's microcomputer. Intent on making it a powerful tool, we have **large scale business accounting systems, data processing systems, system utilities,** and the **Library 100.** We have the only **DISK HEAD CLEANER** (for APPLE too!) and **GRAN MASTER DISKETTES,** the best on the market.

TBS is **YOUR COMPANY,** and we build systems, not just software. The above products are available now, nationwide. Visit your local Computer Dealer or Associate Radio Shack Store and demand the best, demand TBS. For more information, contact us through the numbers below.

™TRS-80 is a trademark of the Tandy Corporation.

TBS™ ✓6

THE BOTTOM SHELF, INC.
(404)939-6031 • P.O. Box 49014 • Atlanta, GA 30359

```
Subroutine Discounted Cash Flow ROI
Low = -1
High = 2
Tol = 0.001
Er = 1
Do while Er>Tel
    ROI = (High + Low)/2
    Sum = 0
    Year = 1
    Do while Year<Life
      Sum = Sum + Cashflow_Year*(1 + ROI) - Yeer
    End do
    If ABS(High - Low)<0.001 then
      print failed to converge
      exit
      Else
      Er = (Sum - Depbase)/Depbase
      If Er>0 Then Low = ROI
        Else High = ROI
      End if
    End if
    Er = ABS(er)
End do
End subroutine
```

Figure 4. Discounted Cash Flow ROI Subroutine.

14. Calculate DSCF ROI using current year dollars.
15. Calculate DSCF ROI using constant year dollars.
16. Print the results.

Each of these 16 steps is a module in the program. Most of them can be understood from the program listing and the remarks in the listing.

Algorithms for the major modules are shown in Figures 1 through 4. These algorithms combined with the TRS-80 Level II BASIC listing should enable you to translate the program to any language that your microcomputer understands.

Data Requirements

Before you run the program, you'll need the following information:

1. Annual revenue: What is your best estimate of the gross annual revenue that the investment will earn?
2. Annual operating cost: What is your best estimate of the total operating cost, excluding interest and depreciation, required by the investment?
3. Escalation rate of revenue: How much do you expect revenue to increase each year, in percent?
4. Escalation rate of costs: What percent do you estimate costs will increase per year?
5. Depreciable base: What is the cost of the investment less non-depreciable items such as land?
6. Salvage value: How much is the depreciable bese worth at the end of the project life?
7. Project life: How long will you keep the investment? This figure is also used for the depreciation time.
8. Amount of loan: How much money do you have to borrow to finance the investmant?
9. Annual interest rate
10. Life of loan
11. Number of payments per year
12. Investment tax credit: Can you take an investment tax credit for this investment and, if so, what percent? Check with the IRS to be sure.
13. Income tax rate: What is the marginal income tax rate you will have to pay on incoma derived from this investment?
14. Gross National Product (GNP) Deflator: What is your estimate of the averaga yearly inflation rate during the life of the project (8 percent is the default)?

Spend some time making sure the estimates are as accurate as you can make them, the accuracy of the program depends on them.

Running the Program

Once you hava assembled the required information, load the program into your computer.

You will be asked to enter each of the items discussed above. Don't worry if you make a mistake. Keep on going. After you've antared all the information it will be displayed on the CRT.

You will be aaked "Are these

Sample Problem

```
                    INPUT DATA
YEARLY REVENUE                                  $ 50000
YEARLY OPERATING COST                           $ 23000
ESCALATION FOR REVENUE                          5 %
ESCALATION FOR COSTS                            5 %
DEPRECIABLE BASE                                $ 65000
SALVAGE VALUE                                   $ 12000
AMOUNT OF LOAN                                  $ 65000
ANNUAL INTEREST RATE                            12 %
LIFE OF LOAN                                    5 YRS
NUMBER OF PAYMENTS/YR                           12
PROJECT LIFE                                    7
INVESTMENT TAX CREDIT RATE                      10 %
INCOME TAX RATE                                 50 %
GNP DEFLATOR                                    8 %
 SIMPLE ROI FOR STAIGHT LINE DEPRECIATION       38.2278 %
 DCCF ROI FOR STRAIGHT LINE DEPRECIATION        22.2412 %
DCCF ROI CORRECTED FOR INFLATION FOR STAIGHT LINE DEPRECIATION 13.1592 %
SIMPLE ROI FOR SUM OF YEARS DIGITS DEPRECIATION 38.2278 %
DCCF ROI FOR SUM OF YEARS DIGITS DEPRECIATION 23.4863 %
DCCF ROI CORRECTED FOR INFLATION FOR SUM OF YEARS DIGITS DEPRECIATION 14.3311 %
SIMPLE ROI FOR DECLINING BALANCE DEPRECIATION 38.2278 %
DCCF ROI FOR DECLINING BALANCE DEPRECIATION  24.292 %
DCCF ROI CORRECTED FOR INFLATION FOR DECLINING BALANCE DEPRECIATION 15.0635 %
```

Sample Listing

```
DETAILS FOR YEAR # 1
REVENUE                                  52500
OPERATING COST                           24150
INTEREST PAID                            7256.75
ANALYSIS FOR STAIGHT LINE DEPRECIATION
DEPRECIATION                             $ 7571.43
PRETAX PROFIT                            $ 13521.8
TAX PAID                                 $$ 268.912
AFTER TAX INCOME                         $ 13260.9
CURRENT YEAR CASH FLOW                   $ 20832.3
DISCOUNTED CURRENT DOLLAR CASH FLOW      $ 17042
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION  $ 15289.2
DISCOUNTED CONSTANT DOLLAR CASH FLOW     $ 17046.1
ANALYSIS FOR SUM OF YEARS DIGITS DEPRECIATION
DEPRECIATION                             $ 13250
PRETAX PROFIT                            $ 7843.25
TAX PAID                                 $$ 0
AFTER TAX INCOME                         $ 7843.25
CURRENT YEAR CASH FLOW                   $ 21093.3
DISCOUNTED CURRENT DOLLAR CASH FLOW      $ 17061.4
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION  $ 15538.8
DISCOUNTED CONSTANT DOLLAR CASH FLOW     $ 17062.7
ANALYSIS FOR DECLINING BALANCE DEPRECIATION
DEPRECIATION                             $ 18571.4
PRETAX PROFIT                            $ 2521.82
TAX PAID                                 $$ 8
AFTER TAX INCOME                         $ 2521.82
CURRENT YEAR CASH FLOW                   $ 21093.3
DISCOUNTED CURRENT DOLLAR CASH FLOW      $ 16976.7
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION  $ 15538.8
DISCOUNTED CONSTANT DOLLAR CASH FLOW     $ 16973.9

DETAILS FOR YEAR # 2
REVENUE                                  55125
OPERATING COST                           25357.5
INTEREST PAID                            5976.55
ANALYSIS FOR STAIGHT LINE DEPRECIATION
DEPRECIATION                             $ 7571.43
PRETAX PROFIT                            $ 16219.5
TAX PAID                                 $$ 8189.76
AFTER TAX INCOME                         $ 8189.76
CURRENT YEAR CASH FLOW                   $ 15681.2
DISCOUNTED CURRENT DOLLAR CASH FLOW      $ 10494.1
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION  $ 13444.1
DISCOUNTED CONSTANT DOLLAR CASH FLOW     $ 10499.1
ANALYSIS FOR SUM OF YEARS DIGITS DEPRECIATION
DEPRECIATION                             $ 11357.1
PRETAX PROFIT                            $ 12433.8
TAX PAID                                 $$ 3638.53
AFTER TAX INCOME                         $ 8795.28
CURRENT YEAR CASH FLOW                   $ 20152.4
DISCOUNTED CURRENT DOLLAR CASH FLOW      $ 13215.7
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION  $ 17277.5
DISCOUNTED CONSTANT DOLLAR CASH FLOW     $ 13217.6
ANALYSIS FOR DECLINING BALANCE DEPRECIATION
DEPRECIATION                             $ 13265.3
PRETAX PROFIT                            $ 10525.7
TAX PAID                                 $$ 23.7363
AFTER TAX INCOME                         $ 10501.9
CURRENT YEAR CASH FLOW                   $ 23767.2
DISCOUNTED CURRENT DOLLAR CASH FLOW      $ 15384.8
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION  $ 20376.6
DISCOUNTED CONSTANT DOLLAR CASH FLOW     $ 15390.6
```

NEW VERSATILITY For Your TRS-80

FROM THE ORIGINATOR OF THE TRS-80 PROJECT

FMG CORPORATION

HEADQUARTERS FOR PASCAL

And Other High Level Languages

PASCAL COMPILER

Pascal/MT™ is designed to run under CP/M. It provides an efficient development cycle plus efficiently executing object programs using a highly structured language so fewer programming errors are made. Pascal/MT has extensions to standard Pascal.

Features:
- ***Compiler executes only in 32K.***
- ***Direct I/O manipulation.***
- ***Logical Functions allow bit manipulation.***
- ***Assembly languages interface.***
- ***Object programs execute ten times faster than P-code system.***
- ***Includes real time symbolic debugger.***
- ***Accurate 16 digit BCD business arithmetic.***

Package Includes System Diskette and Instruction Book (Requires CP/M) price..... **$100.00**

Same package (TRS-DOS version) price..... **$125.00**

Same package (S-100 CPM version with 8" diskette) price..... **$100.00**

HIGHLY RECOMMENDED PASCAL MANUALS

"Pascal Users Manual and Report" Price, Manual Only..... **$9.00**

UCSD Reference Manual Price, Manual Only..... **$25.00**

"Problem Solving Using Pascal," the beginner's book for UCSD Pascal Price, Manual Only..... **$14.00**

"Programming in Pascal" Price, Manual Only..... **$14.00**

"Pascal Primer" Price, Manual Only..... **$17.00**

TEXTWRITER II

New Lower Price

Exclusively from FMG

A text formatting program that prints files created by an editing program. Contracts, personalized form letters and other documents can be printed from a stored library of standard paragraphs.

Price..... **$75.00**
Price (Manual only)..... **$25.00**

TEXTWRITER III

All features of Textwriter II plus ability to computer create Indexes and Table of Contents.

Price..... **$125.00**

CP/M® OPERATING SYSTEM

New 1.5 Version

Includes RS-232 and I/O Byte implementation, Editor, Assembler, Debugger and Utilities for 8080 and Z80 Systems.

Package Includes:
- ***CP/M System Diskette 5¼"***
- ***CP/M Features and facilities Manual***
- ***CP/M Editor's Manual***
- ***CP/M Assembler Manual***
- ***CP/M Debugger Manual***
- ***CP/M Interface Guide***

PRICE..... **$150.00** (Requires 16K and one drive min.)

Price, set of 5 (Manuals only)..... **$25.00**

Update for 1.4 version owners, new disc supplied..... **$20.00**

FORTRAN

New Lower Price

Comparable to compilers on large main frames and minicomputers. All of ANSI standard FORTRAN X 3.9-1966 is included except COMPLEX data type. Therefore, users may take advantage of the many applications programs already written in FORTRAN.

Features:
- ***FORTRAN Compiler***
- ***Macro Assembler (280)***
- ***Linker***
- ***Library***
- ***Lib Manager (Not in TRS-DOS version)***

FORTRAN, TRS-CP/M version price... **$200.00**
FORTRAN, TRS-DOS version price .. **$150.00**
Price for each (Manual only)..... **$25.00**

TRS-80 COMMUNICATOR

New RS232 Communication Program that allows your TRS-80 to transmit or receive programs and data files. Also makes the TRS-80 into a remote terminal. Requires Radio Shack RS232.

TRS-CP/M version..... **$25.00**
TRS-DOS version..... **$35.00**

Call or write for complete information

FMG CORPORATION

(817) 294-2510

P.O. Box 16363(80) • Fort Worth, Texas 76133

✓12

TRS-COBOL-80

FMG's Microsoft COBOL brings the world's most widely used computer programming language to the TRS-80 user. FMG COBOL-80 is comparable to COBOL systems found on minicomputers and large mainframes. Consequently, it greatly enhances the usefulness of microcomputers because it gives users access to the incredibly large numbers of programs already written in COBOL. Because COBOL-80 is a standard, COBOL programs written on other computers may be run easily on the TRS-80.

FMG TRS-COBOL-80 is based on the 1974 ANSI standard. It combines all Level 1 features and the most useful Level 2 options for the "Nucleus" and for sequential Relative and Indexing file handling facilities.

The FMG TRS-COBOL-80 system includes a compiler for translating source code into relocatable object code (which, incidentally, is compatible with the object code of our FORTRAN-80 compiler and MACRO-80 assembler), and a routine system for running the program by interpreting the object code at execution time. It also includes screen handling capabilities for easy use of interaction.

Requires 48K, CP/M and 2 Mini Disk Drives.

Price..... **$750.00**

UCSD PASCAL

The powerful, general purpose language system, originally developed for large, complex system, is now available from FMG for your TRS-80. This new FMG/UCSD Pascal System greatly increases the value and capability of the TRS-80.

Package Includes:
- ***Operating System***
- ***Screen Editor***
- ***280 Macro Assembler***
- ***Library***
- ***Pascal Compiler***
- ***Utilities and System Reference Book***

Price..... **$150.00** (Requires 48K, 2 drive System)

available without Macro Assembler, Linker and Library (not for compiling programs) price..... **$100.00**

For recommended Pascal manuals, refer to "PASCAL COMPILER" listing.

High Level Languages for the TRS-80

BASIC • FORTRAN • COBOL • PASCAL

CP/M is a registered trademark of Digital Research Corp ■ TRS-80 is a registered trademark of Radio Shack

```
DETAILS FOR YEAR # 3
REVENUE                                          57881.3
OPERATING COST                                   26625.4
INTEREST PAID                                    4534.81
ANALYSIS FOR STRAIGHT LINE DEPRECIATION
DEPRECIATION                                     $ 7571.43
PRETAX PROFIT                                    $ 19150.4
TAX PAID                                         $$ 9575.22
AFTER TAX INCOME                                 $ 9575.22
CURRENT YEAR CASH FLOW                           $ 17146.7
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 9386.99
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13611.6
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 9393.74
ANALYSIS FOR SUM OF YEARS DIGITS DEPRECIATION
DEPRECIATION                                     $ 9464.29
PRETAX PROFIT                                    $ 17257.6
TAX PAID                                         $$ 8628.79
AFTER TAX INCOME                                 $ 8628.79
CURRENT YEAR CASH FLOW                           $ 18093.1
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 9688.51
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 14362.9
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 9618.56
ANALYSIS FOR DECLINING BALANCE DEPRECIATION
DEPRECIATION                                     $ 9475.22
PRETAX PROFIT                                    $ 17246.7
TAX PAID                                         $$ 8623.33
AFTER TAX INCOME                                 $ 8623.33
CURRENT YEAR CASH FLOW                           $ 18098.5
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 9475.71
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 14367.2
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 9431.05

DETAILS FOR YEAR # 4
REVENUE                                          68775.3
OPERATING COST                                   27956.7
INTEREST PAID                                    2968.53
ANALYSIS FOR STRAIGHT LINE DEPRECIATION
DEPRECIATION                                     $ 7571.43
PRETAX PROFIT                                    $ 22338.7
TAX PAID                                         $$ 11169.4
AFTER TAX INCOME                                 $ 11169.4
CURRENT YEAR CASH FLOW                           $ 18740.8
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 8393
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13775
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 8481.85
ANALYSIS FOR SUM OF YEARS DIGITS DEPRECIATION
DEPRECIATION                                     $ 7571.43
PRETAX PROFIT                                    $ 22338.7
TAX PAID                                         $$ 11169.4
AFTER TAX INCOME                                 $ 11169.4
CURRENT YEAR CASH FLOW                           $ 18740.8
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 8859.58
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13775
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 8861.88
ANALYSIS FOR DECLINING BALANCE DEPRECIATION
DEPRECIATION                                     $ 6768.81
PRETAX PROFIT                                    $ 23142.1
TAX PAID                                         $$ 11571.1
AFTER TAX INCOME                                 $ 11571.1
CURRENT YEAR CASH FLOW                           $ 18339.1
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 7684.31
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13479.8
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 7696.11

DETAILS FOR YEAR # 5
REVENUE                                          63814.1
OPERATING COST                                   29354.5
INTEREST PAID                                    1876.89
ANALYSIS FOR STRAIGHT LINE DEPRECIATION
DEPRECIATION                                     $ 7571.43
PRETAX PROFIT                                    $ 25811.3
TAX PAID                                         $$ 12905.6
AFTER TAX INCOME                                 $ 12905.6
CURRENT YEAR CASH FLOW                           $ 20477.1
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 7582.84
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13936.4
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 7511.84
ANALYSIS FOR SUM OF YEARS DIGITS DEPRECIATION
DEPRECIATION                                     $ 5678.57
PRETAX PROFIT                                    $ 27784.2
TAX PAID                                         $$ 13892.1
AFTER TAX INCOME                                 $ 13892.1
CURRENT YEAR CASH FLOW                           $ 19538.6
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 6881.77
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13292.2
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 6884.2
ANALYSIS FOR DECLINING BALANCE DEPRECIATION
DEPRECIATION                                     $ 4834.3
PRETAX PROFIT                                    $ 28548.4
TAX PAID                                         $$ 14274.2
AFTER TAX INCOME                                 $ 14274.2
CURRENT YEAR CASH FLOW                           $ 19108.5
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 6441.86
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13884.9
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 6447.94

DETAILS FOR YEAR # 6
REVENUE                                          67884.8
OPERATING COST                                   30822.2
INTEREST PAID                                    8
ANALYSIS FOR STRIGHT LINE DEPRECIATION
DEPRECIATION                                     $ 7571.43
PRETAX PROFIT                                    $ 28611.2
TAX PAID                                         $$ 14305.6
AFTER TAX INCOME                                 $ 14305.6
CURRENT YEAR CASH FLOW                           $ 21877
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 6556.65
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13786.2
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 6566.89
ANALYSIS FOR SUM OF YEARS DIGITS DEPRECIATION
DEPRECIATION                                     $ 3785.71
PRETAX PROFIT                                    $ 32396.9
TAX PAID                                         $$ 16198.4
AFTER TAX INCOME                                 $ 16198.4
CURRENT YEAR CASH FLOW                           $ 19984.2
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 5636.82
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 12593.4
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 5638.43
ANALYSIS FOR DECLINING BALANCE DEPRECIATION
DEPRECIATION                                     $ 85.7383
PRETAX PROFIT                                    $ 36896.9
TAX PAID                                         $$ 18848.4
AFTER TAX INCOME                                 $ 18848.4
CURRENT YEAR CASH FLOW                           $ 18134.2
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 4918.57
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 11427.6
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 4924.14

DETAILS FOR YEAR # 7
REVENUE                                          70355.1
OPERATING COST                                   32363.3
INTEREST PAID                                    8
ANALYSIS FOR STRAIGHT LINE DEPRECIATION
DEPRECIATION                                     $ 7571.43
PRETAX PROFIT                                    $ 30428.3
TAX PAID                                         $$ 15218.2
AFTER TAX INCOME                                 $ 15218.2
CURRENT YEAR CASH FLOW                           $ 22781.6
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 5565.47
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 13292.8
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 5594.86
ANALYSIS FOR SUM OF YEARS DIGITS DEPRECIATION
DEPRECIATION                                     $ 1892.86
PRETAX PROFIT                                    $ 36898.9
TAX PAID                                         $$ 18849.4
AFTER TAX INCOME                                 $ 18849.4
CURRENT YEAR CASH FLOW                           $ 19942.3
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 4554.52
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 11636.1
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 4556.8
ANALYSIS FOR DECLINING BALANCE DEPRECIATION
DEPRECIATION                                     $ 8
PRETAX PROFIT                                    $ 37991.7
TAX PAID                                         $$ 18995.9
AFTER TAX INCOME                                 $ 18995.9
CURRENT YEAR CASH FLOW                           $ 18995.9
DISCOUNTED CURRENT DOLLAR CASH FLOW              $ 4145.31
CONSTANT DOLLAR CASH FLOW FOR  8 % INFLATION     $ 11863.9
DISCOUNTED CONSTANT DOLLAR CASH FLOW             $ 4156.79
```

correct yes or no?" Look the data over carefully. If you see an error, answer "NO." Each of the items will be asked for again and its value displayed in parentheses. If the value in parentheses is correct, press ENTER. If it is not, enter the correct value.

After you have corrected the data, select a depreciation method. If you want to use a particular method, enter the appropriate number, otherwise press ENTER and the program will calculate all three methods—straight line, sum of years digits and declining balance.

Now wait a minute or two. (The longer the life of the investment or of the loan, the longer the calculations take.)

As soon as the calculations are completed, the simple ROI, the DSCF ROI in current year dollars, and DSCF ROI in constant year dollars for each depreciation method will be displayed. If you selected one method of depreciation, only the results for that method will be displayed.

If you want details of the profits, taxes, and cash flow for each year, press ENTER.

After you have seen all the details for each year, you will be asked if you want hard copy. If you answer "YES", all the input data and all the calculated results will be printed.

Closing Comments

First some warnings: The program assumes that all losses can be written off against future profits and that the investment tax credit can be taken regardless of when the profits are taken.

These assumptions are not strictly correct. The IRS limits both tax loss and tax credit carry forward. If you need information on this, contact the IRS.

Finally, as I said before, the numbers calculated by the program are only as good as the information you provided. If you desire a precise answer, you must provide precise input. No matter what the computer says, the final decision is yours.

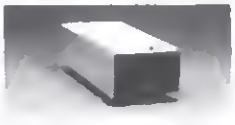
Examine the sample problem. The results are given in the sample listing. ■

An Entire Family of Disk Drives for APPLE, TRS-80*, and S-100 Computers

Only LOBO DRIVES offers you an entire family of fully-compatible disk drives to select from. Whatever computer you're using, APPLE, TRS-80, or S-100, you can add a LOBO drive now, with the peace-of-mind of knowing there's a whole family of drives available when you're ready to expand.

And every drive you order comes complete with chassis and high reliability power supply. Each drive is 100% calibrated, burned-in, and performance tested on either an APPLE, TRS-80, or S-100 computer before it's shipped. We are so proud of our drives . . . our quality, reliability, and performance, that we back-up every drive with a one year, 100% parts/labor warranty.

400 SERIES FLOPPY DISK DRIVES

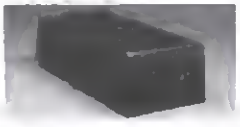


Meet our low-cost 5.25-inch mini drive that records data in either hard or soft sectored format. It is available in single or double density configurations, with a total storage capacity of 220K bytes.

800/801 SERIES FLOPPY DISK DRIVES

Here is our dual 8-inch Floppy disk memory unit. It records and retrieves data on standard 8-inch diskettes to provide 800K bytes of data storage unformatted, or 512K bytes in IBM format per drive. It is also available with double-sided, double-density capabilities, for a maximum storage capacity of 1.6 Megabytes.

7000 SERIES HARD DISK DRIVES



The latest member of our drive family, the Series 7000 is an 8-inch, 10 Megabyte Winchester Technology, hard disk drive. It is fully hardware/software compatible and comes complete with disk controller. Now you can have the convenience, speed, reliability, and all the storage capacity you need.

Call or write for the complete LOBO DRIVES story. Find out just how competitively priced a quality drive can be.

Quantity discounts available –
Dealer inquiries invited.

Yes, I want to know more about LOBO Drives and what they can do. Send me information on:
☐ TRS-80 ☐ APPLE ☐ S-100

☐ 5 1/4-in. Floppy drive
☐ 8-in. Floppy drive
Single sided
Double sided
☐ 8-in. Winchester hard disk, 10 Mbyte drive
☐ Double density expansion interface

Name _____
Company _____
Address _____
City _____ State _____ Zip _____
Phone No. _____
If dealer, provide resale no. _____

* TRS-80 is a registered trademark of Radio Shack, a Tandy Company

LOBO drives
INTERNATIONAL

935 Camino Del Sur
Goleta, California 93017
(805) 685-4546

"CAN YOU REALLY AFFORD TO PAY LESS?"

Use your 80 to fight back. Keep your records in order.

IRS-80

William B. McNeil, Jr.
225 Longview Drive
Jefferson, IN 47130

Do you ever wonder if you are paying too much income tax?

I do, just about every year.

The one thing I always question is the deduction the tex tables allow for the amount I pay in state sales tax. I had to rely on the tax tables year after year because I thought it would be too much trouble to accumulate the necessary records to claim more of a deduction.

In fact, I didn't even know if I could claim more than the tables allowed. But, I thought I should, since it seems—especially in the last few years—that I have been spending more money then I make to meet my daily needs.

The TRS-80 Record Keeper

Here is where my new-found friend, Radio Shack's TRS-80, came to my rescue.

I got my TRS-80 system at the end of last year and started keeping records from the first of January and for each month of the year. I wrote my own programs for this, first in Level I and then, when I upgraded, in Level II BASIC.

I wrote the following program to keep track of the amounts of sales tax that my family pays each day.

The government allows you to use actual sales tax figures, if you can provide proof of the amounts spent.

I'll have my proof for next year's tax return.

From my records I can see that I will be able to claim a substantially greater deduction for sales taxes than the tables allow. In my own case this will be more than double last year's deduction.

My program is simple. It creates a cassette record for each month of the year. A menu of three or four choices is the return point from all functions except End-Of-Job.

The operator is helped through each step for each choice with a prompt, displayed in large character format. Before any tape entry is written to cassette the operator can visually verify its accuracy and re-type the entries if a mistake is noted.

I provided a hard copy output for those who have printers.

The monthly worksheets are updated daily, listing sales taxes paid. At the end of each month the worksheets are entered onto cassette.

At tax time the cassette is printed and retained for your records. If your return is questioned, you will be prepared to back up your figures.

When accumulating your daily records, don't forget that there are many hidden taxes that you can claim such as items purchased from a vending mechine, or meybe the price you pay for gasoline and theater ticketa. Don't forget that your magazine, newspaper, meals away from home, utilities and just about every penny you spend may have sales taxes in the price or added to the price.

Check with your State Department of Internal Revenue to find out just what sales taxes may be included in various purchase prices.

How to Begin

Make yourself twelve of the worksheets similar to the sample I use in Fig. 1. The program allows six sales tax entries per day. A grid six by 31 is adequate.

I use INKEY$ wherever possible to eliminate wrong entries. Also I use CHR$(23) to display a large character for easier reading.

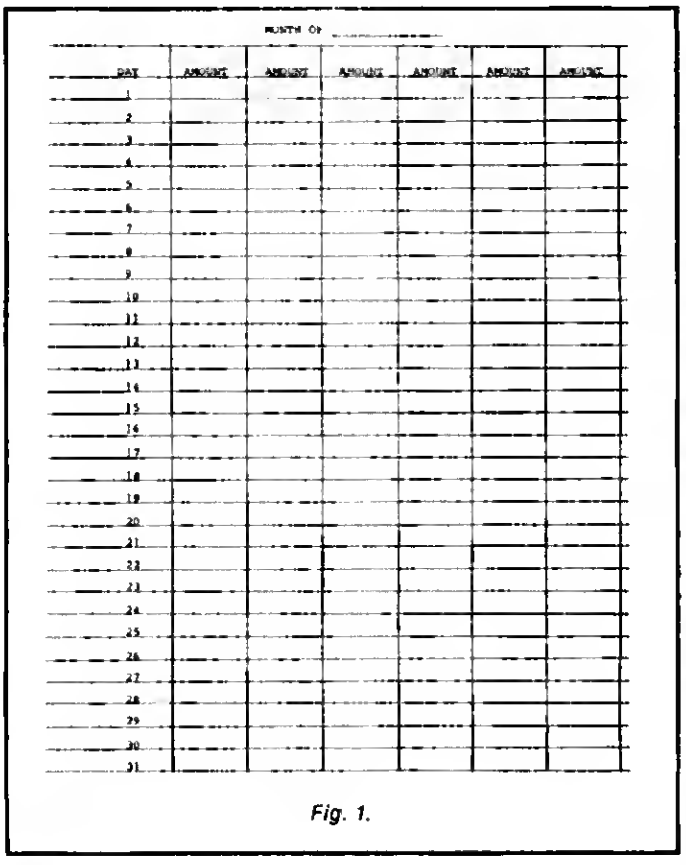
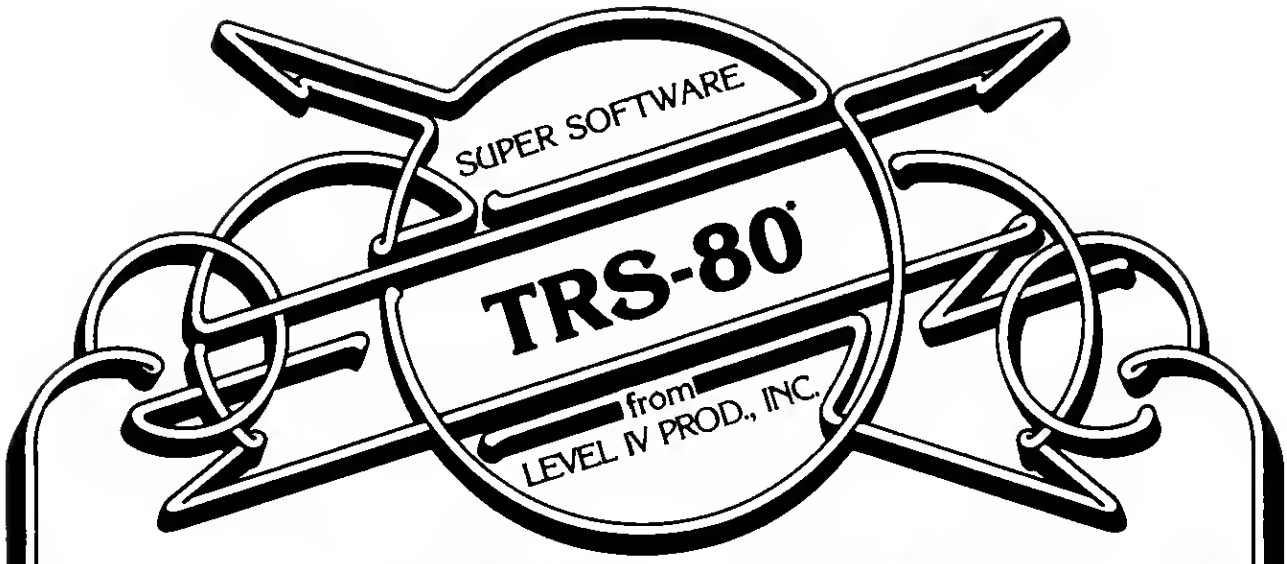


Fig. 1.



Cylon - L IV	$7.50	Payroll Disk - TSE	$59.95	Sargon - Hayd.	19.95
Animated Hangman - L IV	7.50	Checkers Disk - TSE	24.95	Sargon II - Hayd.	29.95
Space Battle CS - L IV	14.95	Personal Finance - TSE	9.95	Pokar Pete - Qual.	9.95
Level IV Graphics CS- L IV	14.95	GSF - RACT	24.95	Fastgammon - Qual.	19.95
Typing I - L IV	15.00	Remodel & Proload - RACT	34.95	Fastgammon Disk - Qual.	19.95
Typing II - L IV	15.00	Remodel - RACT	24.95	Advent Cass - Adve.	14.95
Space Battle Disk - L IV	19.95	Dosort - RACT	34.95	Advent 2 Cass - Adve.	14.95
Typing I & II - L IV	22.50	Comproc - RACT	11.97	Advent 3 Cass - Adve.	14.95
Warfare 1 - TSE	7.95	Infinite Basic - RACT	49.95	Advent 4 Cass - Adve.	14.95
Final Approach - L IV	10.00	Infinite Basic - RACT	29.95	Advent 5 Cass - Adve.	14.95
Cribbage - TSE	7.95	Copy System - RACT	14.95	Advent 6 Cass - Adve.	14.95
Tycoon - TSE	7.95	Autok/Qedit - DISC	14.95	Advent 7 Cass - Adve.	14.95
Concentration - TSE	7.95	Autok/Qedit Sk - Discovery Bay	19.95	Advent 8 Cass - Adve.	14.95
Amazin Maze - TSE	7.95	Morse Code - Discovery Bay	14.95	Advent 10 Cass - Adve.	5.95
Time Bomb - TSE	4.95	Win 21 - Discovery Bay	19.00	Advent 1&2 Disk - Adve.	24.95
9 Games/Pre School - TSE	9.95	Gomoco - Discovery Bay	15.00	Advent 3&4 Disk - Adve.	24.95
Slalom - TSE	7.95	Level 1 - Apparat	15.00	Advent 5,6&7 Disk - Adve.	39.95
Ten Pin - TSE	7.95	New Dos - Apparat	49.00	Advent 8 Disk - Adve.	19.95
Endzone- TSE	7.95	New Dos + - Apparat	99.00	Bowlstats - RLS	29.95
Star Trek 3.4 - TSE	14.95	KVP - L. Micklus	29.95	Forth Disk - Mill.	64.95
Adventure Mission IM - TSE	14.95	KVP-232 - L. Micklus	29.95	Primmer - Mill.	15.00
Dog Star - TSE	9.95	Library 100 - TBS	49.95	Space War - Mill.	9.95
Safari - TSE	7.95	Electric Pencil Cass. - Shay	99.00	Starfleet Orion - Asim.	19.95
Treasure Hunt - TSE	7.95	Electric Pencil Disk - Shay	150.00	Invasion Orion - Asim.	19.95
Round the Horn - TSE	9.95	Snake Eggs/SND - BOUS	14.95	Temple of Apsh - Asim.	24.95
Pork Barrel - TSE	9.95	Android Nim/SND - BOUS	14.95	Misosys Tape U - Acro.	14.95
Kamikaze - TSE	7.95	Life Two/SND - BOUS	14.95	Misosys Disass - Acro.	19.95
ST80D - TSE	79.00	Bee Wary - BOUS	14.95	System Savers - Acro.	14.95
Advance Personal Finance - TSE	24.95	Owl Tree W/S - BOUS	9.95	Opera Theater - Acro.	9.95
Atlantic Balloon - TSE	9.95	Great Race W/S - BOUS	9.95	Disk Utility - Acro.	19.95
Tarot Cards - TSE	9.95	Lying Chimps - BOUS	9.95	Video Pager - Acro.	9.95
Ham Radio - TSE	14.95	Concentration W/S - BOUS	9.95	Tshort - Web.	9.95
Breakaway - TSE	5.95	Scrabble W/S - BOUS	9.95	Wordo - Micr.	14.95
X-Wing Fighter - TSE	9.95	Level III Basic - Micr.	49.95	Stock Exchange - Micr.	15.95
Simple Simon - TSE	4.95	Micropoly - O.P.	7.50	Whereami - Micr	10.95
Sinkum - TSE	4.95	Airaid - SSS	SPECIAL 10.00	Othello - L IV	9.95
Educational Assistant - TSE	9.95	Barricade - SSS	SPECIAL 10.00	Invaders w/sound - L IV	14.95
Home Financial - TSE	9.95	RSM-2 - SSS	26.95	Creature Tic-Tac-Toe	7.50
Math Drill - TSE	4.95	RSM-2D - SSS	29.95	Creature Tic-Tac-Toe w/ sound	9.95
Object System - TSE	9.95	DCV-1 - SSS	9.95	XREF Cass - L. Micklus	19.95
Text 80 Disk - TSE	59.95	Basic P1 - SSS	14.95	Renumber Cass - L. Micklus	7.95

LEVEL IV PRODUCTS, INC. ✓14

32238 Schoolcraft • Livonia, MI 48154 • 313-525-6200 outside Michigan 1-800-521-3305

10% Off All Software Listed

If you don't see what you want call, find out why! We carry nothing but the BEST. We also sell TRS-80* 10% off list price.

We ship C.O.D. - Add $1.50 for shipping and handling **Send 25¢ for Catalog**

* Product of Tandy Corporation

I write the data to cassette after each day's entry, which uses very little memory. This method uses more tape than string manipulation. Still, a full year's information can be written to a 60 minute cassette, and each day's entry is capable of being altered in this format.

When the program is loaded it asks if you have a line printer. If you answer yes, the menu displays a choice of four modes of operation. If you answer no the menu displays only three.

First, create a cassette by selecting choice one and following the instructions.

A monthly total and a year-to-date total is automatically written to cassette as a last record for each month. For the first month entered, the totals are zero, but for additional months the carry-forward totals are read first, using choice two. Just read the last day or two of the last month on the tape.

After the totals are read in, answer the question "is there another month" with no and then select choice one. The carry-forward total will be displayed on the TV.

The day of the month is automatically displayed. If you have no entries for a particular day, key ENTER and the day will advance one number.

Enter the amounts as you would on an adding machine, i.e., don't use decimal points. The entered amount is displayed with the decimal placed by the program.

If there aren't six entries per day, just key ENTER and zeroes are entered by the program for those amounts. Note: The first amount cannot be zero or the day advances.

When the six amounts are entered you have the opportunity to visually verify the amounts. If a mistake is noted, the entire day's amounts can be keyed again before any record is written to the cassette.

If you answer the prompt, "Is

Program Listing

```
1000   REM *BILL MCNEILL, JEFFERSONVILLE, IN. 47130*
1010   REM *SEPTEMBER 12, 1979*
1020   REM *REV. 5.0*
1030   REM *TRS-80 LEVEL II 16K*
1040   GOSUB 9500
1050   REM *TITLE AND PRINTER SETUP*
1060   PRINT@ 384, "THIS IS A SALES TAX TAPE ROUTINE"
1070   S$="$$###.##"
1080   PRINT@ 518,"DO YOU HAVE A LINE PRINTER"
1090   GOSUB 8060
1100   T=0: G1=0: G=0: F=0: GOSUB 9500
1110   PRINT TAB(5) ": : : : M E N U : : : :"
1120   PRINT: PRINT TAB(5) "TO WRITE TO TAPE - - - 1"
1130   PRINT TAB(5) "TO READ THE TAPE - - - 2"
1140   PRINT TAB(5) "TO END THE JOB - - - - 3"
1150   IF P$="Y" GOTO 1180
1160   PRINT@ 590, "ENTER YOUR CHOICE"
1170   GOTO 8040
1180   PRINT TAB(5) "TO PRINT THE TAPE  - - 4"
1190   PRINT@ 590, "ENTER YOUR CHOICE"
1200   GOSUB 8080
1210   IF M$="1" T$=" RECORD "
1220   IF (M$="2") OR (M$="4") T$=" PLAY "
1230   IF M$="4" GOTO 4000
1240   IF M$="3" GOTO 3000
1250   IF M$="2" GOTO 2000
1260   IF M$="1" THEN 1500
1500   REM *TAPE WRITE ROUTINE*
1510   GOSUB 6000: GOSUB 9000
1520   T=0: GOSUB 9500
1530   PRINT@ 384,"TYPE IN MONTH AND YEAR"
1540   PRINT: PRINT
1550   INPUT B$
1560   PRINT@-1, B$
1570   A=1: GOSUB 9500
1580   PRINT "DAY--"A
1590   PRINT
1600   INPUT "AMOUNT--"B
1610   IF B=0 A=A+1: GOSUB 9500: GOTO 1580
1620   B=B/100: PRINT@ 208,"$"; USING S$;B
1630   INPUT "AMOUNT--"C: C=C/100: PRINT@ 272,"$"; USING S$;C
1640   INPUT "AMOUNT--"D: D=D/100: PRINT@ 336,"$"; USING S$;D
1650   INPUT "AMOUNT--"E: E=E/100: PRINT@ 400,"$"; USING S$;E
1660   INPUT "AMOUNT--"F: F=F/100: PRINT@ 464,"$"; USING S$;F
1670   INPUT "AMOUNT--"G: G=G/100: PRINT@ 528,"$"; USING S$;G
1680   PRINT
1690   PRINT "IS THE DATA CORRECT"
1700   GOSUB 9000
1710   IF Q$="Y" GOTO 1740
1720   IF Q$="N" THEN 1730 ELSE 1700
1730   GOSUB 9610: GOSUB 9500: GOTO 1580
1740   T=T+B+C+D+E+F+G
1750   PRINT#-1, A;B;C;D;E;F;G
1760   IF (Q$="Y") AND (B=0) GOTO 2130
1770   GOSUB 9610
1780   PRINT
1790   PRINT "IS THIS THE LAST DAY"
1800   GOSUB 9000
1810   IF Q$="N" A=A+1: GOSUB 9500: GOTO 1580
1820   IF Q$="Y" GOTO 1840
1830   GOTO 1800
1840   G=G1+T: F=T: GOTO 1750
2000   REM *TAPE READ ROUTINE*
2010   GOSUB 6000: GOSUB 9000
2020   GOSUB 9500: INPUT#-1, B$
2030   PRINT@ 20, B$
2040   PRINT
2050   PRINT "DAY"; TAB(10) "A M O U N T S"
2060   FOR X=1 TO 4
2070   A=0: B=0: C=0: D=0: E=0: F=0: G=0
2080   INPUT#-1, A,B,C,D,E,F,G
2090   IF B=0 GOTO 2140
2100   PRINT: PRINT A; TAB(5)  USING S$; B;C;D
2110   PRINT ""; TAB(5)  USING S$; E;F;G
2120   IF X=4 GOTO 7000
2130   NEXT
2140   FOR Z=1 TO 1000
2150   GOSUB 9500
2160   PRINT@ 320, B$;"  TOTAL IS"; USING S$;F
2170   PRINT@ 448, "THE TO-DATE TOTAL IS"; USING S$;G
2180   PRINT: PRINT "PRESS ENTER TO CONTINUE": GOSUB 9000
2190   GOSUB 9500
2200   PRINT@ 394, "IS THERE ANOTHER MONTH "
2210   GOSUB 9000
2220   IF (M$="1") AND (Q$="N") GOTO 1100
2230   IF (M$="1") AND (Q$="Y") G1=G: G=0: F=0: GOTO 1920
2240   IF Q$="N" GOTO 1100
2250   IF Q$="Y" GOTO 2020
2260   GOTO 2210
3000   REM *END-OF-JOB ROUTINE*
3010   GOSUB 9500
3020   PRINT@ 462, "E N D  O F  J O B"
3030   GOTO 3030
4000   REM *PRINTER ROUTINE  (IF PRINTER IS AVAILABLE)*
4010   GOSUB 9500
4020   PRINT "PLACE THE PRINTER ON-LINE": PRINT: PRINT
4030   GOSUB 6030: GOSUB 9000
4040   GOSUB 9500
4050   PRINT@ 384, "DATA IS BEING READ FROM TAPE"
4060   INPUT#-1, B$
4070   LPRINT CHR$(12): FORE 16425,1
4080   LPRINT TAB(27) B$
4090   LPRINT ""
4100   LPRINT TAB(5) "DAY"; TAB(27) "A M O U N T S"
4110   LPRINT ""
4120   INPUT#-1, A,B,C,D,E,F,G
4130   IF B=0 GOTO 4190
4140   IF A<10 Y=8
4150   IF A>9 Y=4
4160   LPRINT TAB(Y)A; USING S$; B,C,D,E,F,G
4170   P=PEEK(16425)
4180   IF P>58 THEN 4070 ELSE 4120
4190   LPRINT ""
4200   LPRINT TAB(5)B$; TAB(10)" TOTAL IS", USING S$;F
4210   LPRINT ""
4220   LPRINT TAB(5) "TO-DATE TOTAL IS", USING S$;G
4230   GOSUB 9500
4240   PRINT@ 384, "IS THERE ANOTHER MONTH TO PRINT"
4250   GOSUB 9000
4260   IF Q$="N" CLS: GOTO 1100
4270   IF Q$="Y" G=0: GOTO 4040
4280   GOTO 4230
5000   REM *ALLOWS 'ENTER' ONLY*
5010   I$="": I$=INKEY$: IF I$="" GOTO 5010
5020   IF ASC(I$)=13 CLS ELSE 5010
5030   RETURN
6000   REM *DISPLAY PROMPTS IN LARGE CHARACTERS*
6010   CLS: PRINT CHR$(23)
6020   IF M$="1" GOTO 6060
6030   PRINT "START TAPE AT BEGINNING --OR--"
6040   PRINT: PRINT
6050   PRINT "READ TAPE TO END OF LAST MONTH"
6060   IF M$<>"1" GOTO 6080
6070   PRINT "TO-DATE TOTAL IS"; USING S$;G1
6080   PRINT: PRINT
6090   PRINT "PLACE RECORDER IN"; T$; "MODE"
6100   PRINT: PRINT
6110   IF M$<>"1" GOTO 6140
6120   PRINT "ERASE FORWARD SLIGHTLY"
6130   PRINT: PRINT
6140   PRINT "PRESS ENTER WHEN READY"
6150   RETURN
7000   REM *TIME DELAY FOR SCREEN DISPLAYS*
7010   FOR I=1 TO 10001: NEXT
7020   GOSUB 9500
7030   PRINT: GOTO 2030
8030   REM *ALLOW Y OR N ONLY*
8010   P$="": P$=INKEY$:  IF P$="" GOTO 8010
8020   IF P$="Y" RETURN
8030   IF P$="N" RETURN ELSE 8010
8040   REM *ALLOW MENU CHOICES 1 OR 2 OR 3 ONLY*
8050   M$=INKEY$:  IF M$="" GOTO 8050
8060   IF (M$="3") OR (M$="2") OR (M$="1") GOTO 1210
8070   GOTO 8050
8080   REM *ALLOW MENU CHOICE 4 ONLY*
8090   M$=INKEY$:  IF M$="" GOTO 8090
8100   IF M$="4" RETURN
8110   IF (P$="Y") AND (M$<"4") GOTO 8080 ELSE 8080
9000   REM *WAIT FOR CHOICE FROM MENU*
9010   Q$="": Q$=INKEY$: IF Q$="" GOTO 9010
9020   RETURN
9500   REM *SET SCREEN TO LARGE CHARACTERS*
9510   CLS
9520   PRINT CHR$(23)
9530   RETURN
9600   REM *ZERO OUT AMOUNT BUFFERS*
9610   B=0: C=0: D=0: E=0: F=0: G=0
9620   RETURN
```

the data correct?", with no, the amounts are zeroed and the same day is ready to be re-keyed.

If you answer the prompt with yes, the data is written to cassette.

The next prompt "is this the last day" will write the monthly total and to-date total to the cassette if you answer yes.

The last record written to cassette has the first tax amount zeroed out. This signals the End of the Month choice two.

The prompt asks "is there another month" before continuing or displeying the menu. If you answer yes, the next month is ready to be entered.

The Cassette Read

The menu choice two is the cassette read. Four amounts are displayed per screen layout before more data is read in. The total amounts are displayed when the last record is read in for the particular month. The prompt "is there another month" allows you to get back to the menu or read in another month from cassette.

Menu choice three terminates the job.

If you have a printer the menu choice four is essentially the same as choice two. Follow the prompts and make sure your printer is on-line.

The TV will display 'data is being read' as the cassette is being played. One month is printed per page for ease of reading. Standard 8 inch by 11 inch paper is utilized.

If you have e printer that uses narrower paper, you may have to modify the program from line 4060 to line 4210 to match your printer requirements.

There you heve it. I hope you will be able to save some of your tax money next year. Maybe you can even set up a service bureau for others. It will take a full year to accumulate your amounts. Stick with it, as I did, and you should see that you may have been paying too much income tax all along. ■

With these programs you can print files while running another program entirely!

SPOOL and DESPOOL

H. S. Gentry
Rt. 1, Box 39B
Earlysville, VA 22936

Simulteneous Peripheral Output OverLap (SPOOL) is a technique used by most large computer systems to prevent program delay because a slow peripheral, like a printer, is not ready. The output data is written (spooled) on a mass storage device and then transferred (despooled) when the peripheral is ready.

Spool

The TRS-80 spooler system is divided into two major sections, SPOOL and DESPOOL. The first of these sections is the output spooler, shown in Listing 1.

The code in line numbers 300 through 440 requests the file name and places it in the device control block (DCB) for the file. Line numbers 470 through 540 open an existing file or create a new one and check for errors. If any error is found, an error message is printed and the spool operation is terminated.

If the file opens without error then lines 550 through 590 connect the spooler to the printer DCB and return control to the operating system.

Now, each time the operating system (DOS, BASIC, etc.) attempts to print a character, the code in lines 650 through 930 is activated. The character is counted and stored in a 256 byte buffer. When this buffer is full it is written to the disk. This procedure continues as long as the user allows it or until an error is detected.

When the spool operation is completed you must close the spool file. This is necessary for two reasons.

First, the data printed may not have ended on a 256 byte boundary. Thus, some data may be in the buffer that has not been written to the file. Closing SPOOL will detect this situation, set the unused area of the buffer to zeroes and write the last buffer to the file.

The second reason is that the system program CLOSE must be called to update the disk directory.

The spool system performs both of these close operations, if control is transferred to label KLOSE (location FE76H in Listing 1). This may be done by entering DEBUG and typing GFE76. The memory containing the KLOSE program, the file DCB, the pointers and the 256 byte buffer must not be changed

Program Listing 1.

```
                00100 ;THIS IS THE PRINTER SPOOLER - WHEN LOADED
                00110 ;IT WILL INTERCEPT ALL PRINTER OUTPUT AND
                00120 ;STORE IT IN A 256 BYTE BUFFER WHEN THE
                00130 ;BUFFER IS FULL THE DATA IS WRITTEN TO
                00140 ;THE SPECIFIED FILE.  THE SPOOL FILE MUST
                00150 ;BE CLOSED BY RUNNING THE SYSTEM PROGRAM
                00160 ;CLOSE.
                00170 ;
4467            00180 DISP    EQU     4467H
0040            00190 INPUT   EQU     40H
4026            00220 PRDD    EQU     4026H
402D            00230 DOS     EQU     402DH
4428            00240 CLOSE   EQU     4428H
4420            00250 INIT    EQU     4420H
443C            00270 WRITE   EQU     443CH
                00280 ;
FE00            00290         ORG     0FE00H
FE00 21C5FE     00300 SETUP   LD      HL,MSG1         ;LOD ON
FE03 CD6744     00310         CALL    DISP
FE06 21A4FE     00320         LD      HL,INBFR
FE09 0620       00330         LD      B,32
FE0B CD4000     00340         CALL    INPUT
FE0E 78         00350         LD      A,B             ;GET ACTUAL #
FE0F B7         00360         OR      A
```

until the close operation is done.

If you don't like using DEBUG to close your file you can create a close program as follows: load (but don't execute) the SPOOL program, then dump the KLOSE part of SPOOL to a disk file called CLOSE/CMD. Don't dump more memory than needed. Actually, you only need an execution (transfer) address.

The dump command to close the file for the SPOOL in Program Listing 1 is: DUMP CLOSE/CMD:0 (START = X'FE76',END = X'FE9D',TRA = X'FE76').

Now, after your spool operation is finished, return to DOS and type CLOSE. The file is then closed and the spool operation termineted. You are left with an ASCII file containing all the printer output since the spool was started.

Despool

If you want to print a copy of the spool file the command PRINT could be used. However, this ties up the system while the printer is running.

Fortunately, there is a better way, DSPOOL, shown in Listing 2. This program opens the spool file for printing and returns to the operating system.

The data in the file is then printed while you perform almost any other job on your system. That's right, you can run a BASIC program or perform other disk operations while the file is being printed.

There are only a few exceptions: You cannot re-boot the system; You cannot write to the spool file while despooling; You cannot print data in the regular DOS manner until the despool is completed; You cannot spool on file while despooling another.

The last restriction is included only because SPOOL and DSPOOL use the same memory.

If you move one of the programs to another location, you could SPOOL and DSPOOL at the same time, although you still may not write and read the same file at one time. You must use two different file names.

DSPOOL uses two links to the operating system, one to the 25 millisecond interrupt and anoth-

```
FE10 28EE      00370          JR      Z,SETUP          ;NO INPUT
FE12 E8        00380          EX      DE,HL
FE13 83        00390          ADD     A,E              ;ADDRESS+8
FE14 6F        00400          LD      L,A              ;LOW ADDRESS
FE15 7A        00410          LD      A,D              ;HI ADD
FE16 CE00      00420          ADC     A,0
FE18 67        00430          LD      H,A              ;HI ADDRESS
FE19 3620      00440          LD      (HL),20H         ;BLANK CR
               00450 ;INBFR NOW HAS FILE SPEC WITH TRAILING BLANKS
               00460 ;INIT THE FILE
FE1B 21E1FE    00470          LD      HL,BUFFER        ;PLACE
FE1E 11A4FE    00480          LD      DE,INBFR         ;DCB
FE21 0600      00490          LD      B,0
FE23 CD2044    00500          CALL    INIT             ;OPEN IT
FE26 2809      00510          JR      Z,OK             ;Z=1 IF OK
FE28 21D5FE    00520          LD      HL,ERM
FE2B CD6744    00530          CALL    DISP
FE2E C32D40    00540          JP      DOS              ;AND GET OUT
FE31 2A2640    00550 OK       LD      HL,(PRDD)        ;OLD DRIVER
FE34 22A2FE    00560          LD      (SAVDD),HL       ;SAVE IT
FE37 2140FE    00570          LD      HL,DRIVE         ;NEW DRIVER
FE3A 222640    00580          LD      (PRDD),HL        ;PUT IT IN
FE3D C32D40    00590          JP      DOS              ;DONE
               00600 ;FILE IS OPEN - THIS IS THE ACTUAL DRIVER
               00610 ;IT WILL STUFF THE CHARACTERS IN THE BUFFER
               00620 ;IF THE BUFFER IS FULL A WRITE TO THE DISK
               00630 ;WILL BE DONE.
               00640 ;
FE40 E5        00650 DRIVE    PUSH    HL
FE41 F5        00660          PUSH    AF
FE42 2A9EFE    00670          LD      HL,(PRT)         ;POINT TO BUFFER
FE45 71        00680          LD      (HL),C           ;SAVE CHARACTER
FE46 23        00690          INC     HL
FE47 229EFE    00700          LD      (PRT),HL
FE4A 3AA0FE    00710          LD      A,(CCNT)         ;COUNT
FE4D FEFF      00720          CP      0FFH             ;DUN
FE4F 2807      00730          JR      Z,OUT
FE51 3C        00740          INC     A                ;COUNT IT
FE52 32A0FE    00750          LD      (CCNT),A         ;PUT IT BACK
FE55 F1        00760 POP      POP     AF
FE56 E1        00770          POP     HL
FE57 C9        00780          RET                      ;GO BACK
FE58 C5        00790 OUT      PUSH    BC
FE59 D5        00800          PUSH    DE
FE5A DDE5      00810          PUSH    IX
FE5C FDE5      00820          PUSH    IY
FE5E 11A4FE    00830          LD      DE,INBFR         ;DCB
FE61 CD3C44    00840          CALL    WRITE
FE64 21E1FE    00850          LD      HL,BUFFER
FE67 229EFE    00860          LD      (PRT),HL         ;RESTORE POINTER
FE6A AF        00870          XOR     A                ;A=0
FE6B 32A0FE    00880          LD      (CCNT),A
FE6E FDE1      00890          POP     IY
FE70 DDE1      00900          POP     IX
FE72 D1        00910          POP     DE
FE73 C1        00920          POP     BC
FE74 18DF      00930          JR      POP
               00940 ;THIS IS THE CLOSE ROUTINE - CALLED BY
               00950 ;THE CLOSE FUNCTION TO CLOSE OUT THE LAST
               00960 ;RECORD AND THEN CLOSE THE FILE
FE76 3AA0FE    00970 KLOSE    LD      A,(CCNT)         ;COUNT
FE79 B7        00980          OR      A
FE7A 2813      00990          JR      Z,KLOS           ;NO DATA CLOSE FILE
               01000 ;DATA IN FILE - NULL REMMAINDER THEN WRITE AND CLOSE
FE7C 2A9EFE    01010          LD      HL,(PRT)
FE7F 3600      01020 LOPC     LD      (HL),0
FE81 FEFF      01030          CP      0FFH             ;DUN
FE83 2804      01040          JR      Z,WRIT           ;FULL WRIT IT
FE85 3C        01050          INC     A
FE86 23        01060          INC     HL
FE87 18F6      01070          JR      LOPC
               01080 ;THIS IS THE WRIT TO THE DISK ROUTINE
FE89 11A4FE    01090 WRIT     LD      DE,INBFR         ;DCB
FE8C CD3C44    01100          CALL    WRITE
               01110 ;THIS IS THE CLOSE ROUTINE - IT WILL CLOSE THE
               01120 ;FILE
FE8F 11A4FE    01130 KLOS     LD      DE,INBFR         ;DCB
FE92 CD2844    01140          CALL    CLOSE
FE95 2AA2FE    01150          LD      HL,(SAVDD)
FE98 222640    01160          LD      (PRDD),HL        ;RESTORE PRINTER
FE9B C32D40    01170          JP      DOS              ;DONE
FE9E E1FE      01180 PRT      DEFW    BUFFER
FEA0 0000      01190 CCNT     DEFW    0
FEA2 0000      01200 SAVDD    DEFW    0
FEA4 20        01210 INBFR    DEFM    '
FEC5 53        01220 MSG1     DEFM    'SPOOL FILESPEC?'
FED4 03        01230          DEFB    3
```

```
FED5 53        01240 ERM     DEFM    'SPOOL ERROR'
FEE0 03        01250         DEFB    3
FEE1 00        01260 BUFFER  DEFB    0
FE00           01270         END     SETUP
00000 TOTAL ERRORS
28521  TEXT AREA BYTES LEFT


BUFFER FEE1 01260   00470 00850 01180
CCNT   FEA0 01190   00710 00750 00880 00970
CLOSE  4428 00240   01140
DISP   4467 00180   00310 00530
DOS    402D 00230   00540 00590 01170
DRIVE  FE40 00650   00570
ERM    FED5 01240   00520
INBFR  FEA4 01210   00320 00480 00830 01090 01130
INIT   4420 00250   00500
INPUT  0040 00190   00340
KLOS   FE8F 01130   00990
KLOSE  FE76 00970
LOPC   FE7F 01020   01070
MSG1   FEC5 01220   00300
OK     FE31 00550   00510
OUT    FE58 00790   00730
POP    FE55 00760   00930
PRDD   4026 00220   00550 00580 01160
PRT    FE9E 01180   00670 00700 00860 01010
SAVDD  FEA2 01200   00560 01150
SETUP  FE00 00300   00370 01270
WRIT   FE89 01090   01040
WRITE  443C 00270   00840 01100
```

Program Listing 2.

```
                00110 ;PRINTER DE-SPOOLER - WHEN LOADED IT CONNECTS
                00120 ;TO THE 25MS INTERRUPT AND TO THE KEYBOARD
                00130 ;SCAN ROUTINE. THE SPECIFIED FILE WILL BE
                00140 ;LOADED ONE RECORD AT A TIME INTO LOCAL BUFFER
                00150 ;AND THE INTERRUPT HANDLER WILL PRINT ONE
                00160 ;CHARACTER EACH TIME THE PRINTER IS READY.
                00170 ;WHEN THE EOF IS FOUND THE LINK TO THE
                00180 ;INTERRUPT HANDLER AND THE KEYBOARD SCAN
                00190 ;IS REMOVED.
                00200 ;
4467            00210 DISP    EQU     4467H           ;DISPLAY MESSAGE
0040            00220 INPUT   EQU     40H             ;INPUT MESSAGE
4424            00230 OPEN    EQU     4424H           ;OPEN A FILE
4436            00240 READ    EQU     4436H           ;READ A FILE
4510            00250 MS25    EQU     4510H           ;25 MS QUEUE
4016            00260 KBDD    EQU     4016H           ;POINTER TO KEYBOARD
402D            00280 DOS     EQU     402DH           ;RTN TO DOS
00EA            00290 CNTREG  EQU     0EAH            ;CONTROL/STAT UART
00E8            00300 DTAREG  EQU     0E8H            ;DATA
3FFF            00320 ALIV    EQU     3FFFH
00E8            00330 RESURT  EQU     0E8H
00E9            00340 SWITCH  EQU     0E9H
                00350 ;
FD00            00360         ORG     0FD00H
FD00 D3E8       00370 SETUP   OUT     (RESURT),A      ;RESET UART
FD02 DBE9       00380         IN      A,(SWITCH)      ;READ SWITCHES
FD04 E6F8       00390         AND     0F8H            ;KILL LOW THREE
FD06 F604       00400         OR      04H
FD08 D3EA       00410         OUT     (CNTREG),A
FD0A DBE9       00420         IN      A,(SWITCH)
FD0C E607       00430         AND     07H
FD0E 2172FD     00440         LD      HL,BDTABL
FD11 0600       00450         LD      B,0
FD13 4F         00460         LD      C,A
FD14 09         00470         ADD     HL,BC
FD15 7E         00480         LD      A,(HL)
FD16 D3E9       00490         OUT     (SWITCH),A
                00500 ;UART IS SETUP NOW TALK TO OPERATOR
FD18 2141FE     00510         LD      HL,MSG1
FD1B CD6744     00520         CALL    DISP
FD1E 211AFE     00530         LD      HL,INBFR
FD21 0620       00540         LD      B,32
FD23 CD4000     00550         CALL    INPUT
FD26 78         00560         LD      A,B             ;GET ACTUAL #
FD27 B7         00570         OR      A
FD28 28D6       00580         JR      Z,SETUP         ;NO INPUT
FD2A EB         00590         EX      DE,HL
FD2B 83         00600         ADD     A,E             ;ADDRESS+#
FD2C 6F         00610         LD      L,A             ;LOW ADDRESS
FD2D 7A         00620         LD      A,D             ;HI ADD
```

er to the keyboard driver.

The TRS-80 hardware interrupts the microcomputer forty times per second. The operating system uses this interrupt to run foreground tasks. These tasks include the real time clock, TRACE, or any job you'd like to run.

To run a given job you need to store the address of a pointer in the 25 millisecond queue list. The queue list is at memory location 4510H and 4511H. The pointer is two memory bytes containing the address of your program.

This is a little confusing so let's look at Listing 2 to see what it means.

Lines 800 through 850 put the address of something called PINT in locations 4510H and 4511H. Notice that the code also saves the former contents of 4510H, 4511H to be put back later. PINT is a pointer that contains the memory address of your program.

In this example, 4510H, 4511H contains FD7A (the address of PINT) and FD7AH contains the address of INTHDL (FD7CH). Now, every 25 milliseconds INTHDL, the interrupt handler, is run.

INTHDL

The function of the DSPOOL interrupt handler INTHDL is very simple. It checks the RS232 board to see if it will accept an output character. If the RS232 board is not ready, INTHDL returns to the operating system. If a character can be output, INTHDL checks CCNT.

As long as CCNT is zero, INTHDL returns to the system. If it isn't, one character is output and counted. If the character is a carriage return, the buffer is set up to output a line feed. As long as there is data in the buffer, INTHDL will print it. All of this takes place in time stolen from your other work by the interrupt.

Getting data to the buffer is SCAN's job. SCAN reads one record every time the print buffer is empty (CCNT = 0). It is linked to the TRS-80 keyboard driver and runs every time the system checks the keyboard for input.

```
FD2E CE00      00630          ADC      A,0
FD30 67        00640          LD       H,A              ;HI ADDRESS
FD31 3620      00650          LD       (HL),20H         ;BLANK CR
               00660 ;INBFR NOW HAS FILE SPEC WITH TRAILING BLANKS
               00670 ;INTERRUPT DRIVER IS LINKED ANY TIME CCNT IS
               00680 ;NOT ZERO IT WILL PUT OUT THE NEXT CHARACTER
               00690 ;
               00700 ;NOW TIME TO OPEN THE SPOOL FILE
FD33 2161FE    00710          LD       HL,BUFFER        ;PLACE TO PUT DATA
FD36 111AFE    00720          LD       DE,INBFR         ;DCB
FD39 0600      00730          LD       B,0              ;LRL=0
FD3B CD2444    00740          CALL     OPEN
FD3E 2809      00750          JR       Z,OK             ;Z=1 IF OK
FD40 2152FE    00760          LD       HL,ERM
FD43 CD6744    00770          CALL     DISP
FD46 C32D40    00780          JP       DOS              ;AND GET OUT
               00790 ;LINK 25 MS DRIVER
FD49 F3        00800 OK       DI
FD4A 2A1045    00810          LD       HL,(M525)        ;OLD ONE
FD4D 225FFE    00820          LD       (SAV25),HL       ;SAVE IT
FD50 217AFD    00830          LD       HL,PINT          ;PDINTER
FD53 221045    00840          LD       (M525),HL        ;LINK
FD56 FB        00850          EI
FD57 2A26FE    00860          LD       HL,(SEC)         ;GET SECTORS
FD5A 2217FE    00870          LD       (SECTOR),HL
FD5D 3A22FE    00880          LD       A,(BX)           ;GET BYTES TO EOF
FD60 3219FE    00890          LD       (BCNT),A
               00900 ;FILE OPEN OK NOW LINK KBD SCAN AND GET OUT
               00910 ;KBD SCAN WILL THEN FIND BUFFER EMPTY
               00920 ;AND READ A RECORD.
FD63 2A1640    00930          LD       HL,(KBDD)        ;GET OLD ADDRESS
FD66 22C9FD    00940          LD       (KEY),HL         ;SAVE FOR CONTINUE
FD69 2189FD    00950          LD       HL,SCAN          ;NEW SCAN
FD6C 221640    00960          LD       (KBDD),HL        ;LINKED
               00970 ;SCAN IS NOW LINKED. NEED ONLY TO ENABLE
               00980 ;INTERRUPTS AND GET BACK TO DOS.  SCAN WILL
               00990 ;BE RUN EVERY TIME KEYBOARD IS CHECKED
               01000 ;INTHDL WILL BE RUN EVERY 25 MS
FD6F C32D40    01010          JP       DOS              ;GET OUT
               01020 ;THIS IS THE BAUDE RATE TABLE
FD72 22        01030 BDTABL   DEFB     22H
FD73 44        01040          DEFB     44H
FD74 55        01050          DEFB     55H
FD75 66        01060          DEFB     66H
FD76 77        01070          DEFB     77H
FD77 AA        01080          DEFB     0AAH
FD78 CC        01090          DEFB     0CCH
FD79 EE        01100          DEFB     0EEH
               01110 ;
               01120 ;THIS IS INTHDL THE INTERRUPT HANDLER
               01130 ;IT WILL PRINT A CHARACTER IF CCNT IS NOT
               01140 ;ZERO AND THE PRINTER IS READY.
FD7A 7CFD      01150 PINT     DEFW     INTHDL           ;PDINTER TO INTHDL
FD7C F5        01160 INTHDL   PUSH     AF               ;SAVE AF
FD7D E5        01170          PUSH     HL
FD7E 3AFF3F    01180          LD       A,(ALIV)
FD81 3C        01190          INC      A
FD82 32FF3F    01200          LD       (ALIV),A
FD85 DBEA      01210          IN       A,(CNTREG)       ;STATUS
FD87 CB77      01220          BIT      6,A              ;READY
FD89 281F      01230          JR       Z,CONT           ;NOPE GO ON
FD8B 2A3FFE    01240          LD       HL,(CCNT)        ;CHAR COUNT
FD8E 7D        01250          LD       A,L
FD8F FE00      01260          CP       0
FD91 2005      01270          JR       NZ,OTPT          ;PUT IT OUT
FD93 7C        01280          LD       A,H              ;L=0 CHECK H
FD94 FE00      01290          CP       0
FD96 2812      01300          JR       Z,CONT           ;ALL ZERO GET OUT
FD98 2B        01310 OTPT     DEC      HL               ;-1
FD99 223FFE    01320          LD       (CCNT),HL        ;PUT IT BACK
FD9C 2A3DFE    01330          LD       HL,(ADDR)        ;GET ADDRESS OF CHAR
FD9F 7E        01340          LD       A,(HL)           ;DATA
FDA0 D3EB      01350          OUT      (DTAREG),A       ;OUTPUT IT
FDA2 FE0D      01360          CP       0DH              ;IS IT CR?
FDA4 2807      01370          JR       Z,CR             ;YES
FDA6 23        01380          INC      HL               ;BUMP ADDRESS
FDA7 223DFE    01390          LD       (ADDR),HL
FDAA E1        01400 CONT     POP      HL
FDAB F1        01410          POP      AF
FDAC C9        01420          RET                       ;DONE GET OUT
               01430 ;FOUND CR INSERT LF
FDAD 3E0A      01440 CR       LD       A,0AH            ;CR
FDAF 77        01450 XCR      LD       (HL),A           ;PUT IN BUFFER
               01460 ;AND DONT BUMP ADDRESS.
FDB0 2A3FFE    01470          LD       HL,(CCNT)        ;GET COUNT BACK
FDB3 23        01480          INC      HL               ;PUT 1 IN FOR LF
FDB4 223FFE    01490          LD       (CCNT),HL        ;PUT IT BACK
FDB7 18F1      01500          JR       CONT             ;DO ON
               01510 ;
               01520 ;THIS IS SCAN - IT IS LINKED TO KEYBOARD SCAN
```

If there is data in the buffer, SCAN returns control to the keyboard driver. But, if the buffer is empty, SCAN performs a file read, delaying the keyboard input for about one second.

If all the data has been read from the file, SCAN disconnects the DSPOOL program.

If your printer is 110 baud, the disk reads occur about every 30 seconds. The spool system does not drive any printer faster than 40 characters per second (one per interrupt).

If your printer is faster than this, it will slow down to 40 CPS. At 40 CPS the disk reads occur about every 7.5 seconds. If reading at this rate interferes with the keyboard too much, then add a counter to INTHDL to slow the printer and thus the reads.

Another technique that reduces disk reads is reading two (or more) sectors at a time. However, this complicates the procedure used to find the end of the data.

Modifications

The DSPOOL program shown in Listing 2 is for a serial printer using the Radio Shack RS232 board. The program can be used with a parallel printer (such as the standard printers sold by Radio Shack) by making a few changes.

Delete lines 370 through 500 and move the label SETUP to line 510. Replace lines 1210 through 1230 with the code in Listing 3. Replace line 1350 with LD (37E8H),A.

If your printer automatically feeds a line on every carriage return then delete lines 1360 through 1370 and lines 1430 through 1500.

If you use SPOOL-DSPOOL with NEWDOS or NEWDOS 30, it works as is. If you use it with TRSDOS 2.1, TRSDOS 2.2 or VTOS 3.0, you must add DEC HL between lines 860 and 870. This is necessary because the NEWDOS DCB maintains the number of sectors in a file, while the other systems maintain the number of sectors plus one.

If you use TRSDOS 2.2, change the program ORG and move both programs down to al-

TRS-80 USERS

Natural language is here!

LEVEL 2 BASIC

Can your computer read and solve this problem by itself?

"ON THEIR VACATION, TOM AND DICK VISITED A FARM. WHILE THERE, THEY NOTICED A PEN CONTAINING CHICKENS AND PIGS. TOM SAID THERE WERE 3 TIMES AS MANY CHICKENS AS PIGS. DICK SAID HE COUNTED 100 LEGS IN THE PEN. HOW MANY CHICKENS WERE IN THE PEN?"

with NLOS/1, it can!

NLOS/1-16K; NLOS/2-32K, LARGER VOCABULARY, FASTER, ACTION VERBS – CREATES AND RUNS ITS OWN BASIC PROGRAMS – LEARNS PROCEDURES!

STOP PROGRAMMING YOUR COMPUTER, EDUCATE IT! ORDER TODAY!

SOURCE $1.95 | CASSETTE $4.95

CYBERMATE

R.D. #3 BOX 192A
NAZARETH PA 18064

THE ULTIMATE TRS-80 SPEED-UP!

Mumford Micro Systems announces the release of the SK-2. The most versatile clock modification for the TRS-80 available. It features three speeds: normal (1.77MHz), 50% faster, or 50% slower; selectable at any time without interrupting execution or crashing the program. It may be configured by the user to change speed with a toggle switch or on software command. It may be tied to the expansion interface and will automatically return to normal speed anytime a disk drive is active. It even has provisions for adding an LED to indicate when the computer is not at the normal speed. It mounts inside the keyboard unit with only 4 necessary connections for the switch option (switch not included), and is easily removed if the computer ever needs service. The SK-2 comes fully assembled with illustrated instructions for implementing the various options and complete satisfaction is guaranteed....$24.95

DUPLICATE SYSTEM TAPES WITH "CLONE"

This machine language program makes duplicate copies of ANY tape written for Level II. They may be SYSTEM tapes (continuous or not) or data lists. It is not necessary to know the file name or where it loads in memory, and there is no chance of system co-residency. The file name, entry point, and every byte (in ASCII format) are displayed on the video screen. Data may be modified before copy is produced. CLONE....$16.95

RAM TEST FOR LEVEL II

This machine language program tests memory chips for open or shorted address or data lines as well as intermittents. It tests each BIT for validity and each BYTE in the execution of an actual instruction as in real program execution. Bad addresses are displayed along with the bad data and proper data. One complete test of 48K takes just 14 seconds. Also includes a test for errors induced by power line glitches from external equipment. RAMTEST ...$9.95

PROGRAM INDEX FOR DISK BASIC

Assemble an alphabetized index of your entire program library from disk directories. Program names and free space are read automatically (need not be typed in) and may be alphabetized by disk or program. The list may also be searched for any disk, program, or extension, disks or programs added or deleted, and the whole list or any part sent to the printer. Finally, the list itself may be stored on disk for future access and update. One drive and 32K required. INDEX....$19.95

EDIT BASIC PROGRAMS WITH ELECTRIC PENCIL

This program allows disk users to load Basic programs or any other ASCII data file into the disk version of Electric Pencil for editing. Now you can edit line numbers, move program segments, duplicate program segments, and search for the occurance of any group of characters. PENPATCH....$9.95

SPOOLER FOR PARALLEL PRINTERS ✓144

This program is a full feature print formatting package featuring user definable line and page length (with line feeds inserted between words or after punctuation), screen dump, keyboard debounce, and printer pause control. In addition, printing is done from a 4K buffer area so that the LPRINT or LLIST command returns control to the user while printing is being done. Ideal for Selectric or other slow printers. Allows printing and processing to run concurrently. SPOOLER....$16.95

Include 75¢ postage. California residents add 6% sales tax.
Complete satisfaction or full refund

MUMFORD MICRO SYSTEMS
BOX 435-E Summerland, California 93067 (805) 969-4557

SIMUTEK PRESENTS

★ TRS-80 ★

GAMES

!!! WHOLESALE !!!

★ PACKAGE ONE ★

GRAPHIC-TREK "2000" — This full graphics, real time game is full of fast, exciting action! Expanding photon torpedoes and phasers fill the screen! You must actually navigate the enterprise to dock with the silent space stations as well as to avoid klingon torpedoes! Has shields, galactic memory readout, damage reports, long range sensors, etc! Has 3 levels for beginning, average, or expert players! ★ **INVASION WORM** — Time: 3099, Place: Earth's Solar System Mission: As general of Earth's forces, your job is to stop the Worm invasion and destroy their outposts on Mars, Venus, Saturn, Neptune, etc! Earth's Forces: Androids — Space Fighters — Lazer Cannons — Neutrino Blasters! Worm Forces: Robots — Saucers — Disintegrators — Proton Destroyers! Multi horn gated lets you advance to a more complicated game as you get better! ★ **STAR WARS** — Maneuver your space fighter deep into the nucleus of the Death Star! Drop your bomb, then escape via the only exit. This graphics game is really fun! May the Force be with you! ★ **SPACE TARGET** — Phaser at enemy ships with your missiles. If they react as a paranoiac, capture them — or if you're cruel, destroy them! Full graphics, real time game! ★ **SAUCERS** — This fast action graphics game has a level limit! Can you be the commander to win the distinguished crown! Requires split second timing to play! Watch out!

★ PACKAGE TWO ★

CHECKERS 2.1 — Finally! A checkers program that will challenge everyone! Expert as well as amateur! Uses 3-ply tree search to find best possible move. Picks randomly between equal moves to assure you of never having identical games. ★ **POKER FACE** — The computer uses psychology as well as logic to try and beat you at poker. Cards are displayed using TRS-80's full graphics. Computer raises, calls, and sometimes even folds! Great practice for your Saturday night poker match! (Plays 5 card draw). ★ **PSYCHIC** — Tell me a compulsion a little about yourself and he'll predict things about you, you won't believe! A real mind bender! Great amusement for parties. ★ **TANGLE MANIA** — Try and force your opponent into an immobile position. But watch out, they're doing the same to you! This graphics game is for 2 people and has been used to end several arguments. (And occasionally starts them!) ★ **WORD SCRAMBLE** — This game is for two or more people. One person inputs a word to the computer while the others turn away. The computer scrambles the word, then spews trash of wrong guesses.

★ PACKAGE THREE ★

POETRY — This program lets you choose the subject as well as the mood of the poem you want. You give TRS-80 certain nouns or names, tells the mood, and it does the rest! It has a 1000-word + vocabulary of nouns, verbs, adjectives and adverbs! ★ **ELECTRIC ARTIST** — Manual: draw, erase, move as well as Auto: draw, erase and move. Uses graphics bits not bytes. Saves drawing on tape or disk. ★ **GALACTIC BATTLE** — The Swinsus enemy have long range phasers but cannot travel at warp speed! You can, but only have short range phasers! Can you blitzkrieg the enemy without getting destroyed! Full graphics — real time! ★ **WORD MANIA** — Can you guess the computer's words using your human intuitive and logical abilities? You'll need to, to beat the computer! ★ **AIR COMMAND** — Battle the Kamikaze shots. Requires split second timing. This is a FAST action arcade game.

★ PACKAGE FOUR ★

LIFE — This Z-80 machine language program uses full graphics! Over 100 generations per minute make it truly animated! You make your starting pattern, the computer does the rest! Program can be stopped and changes made! Watch it grow! ★ **SPACE LANDER** — This full graphics simulator lets you pick what planet, asteroid or moon you wish to land on! Has 3 skill levels that make it fun for everyone. ★ **GREED II** — Multi-level game is fun and challenging! Beat the computer at this dice game using your knowledge of odds and luck! Computer keeps track of all winnings and yours. Quick fast action. This game is addictive! ★ **THE PHARAOH** — Build the ancient city of Akhenatai! Buy or lease land. Keep your people from revolting! Steal the rampaging rats. Requires a true political personality to become good! ★ **ROBOT HUNTER** — A group of renegade robots have escaped and are spotted in an old ghost town on Mars! Your job as "Robot Hunter" is to destroy the pirate machines before they kill any more settlers! Exciting! Challenging! Full graphics!

★ PACKAGE FIVE ★

SUPER HORSERACE — Make your bets just like at the real racetrack! 8 horses race in this spectacular graphic display! Up to 9 people can play! Does not deal out but has that element of chance you see in real life! Keeps track of everyone's winnings and losses. This is one of the few computer simulations that can actually get a room of people cheering! ★ **MAZE MOUSE** — The mouse with a mind! The computer generates random mazes of whatever size you specify, level matched for a way out! The second time, he'll always go fastest route! A true display of artificial intelligence! Full graphics, mazes & mouse! ★ **AMOEBA KILLER** — You command a one room submarine that has been shrunken to the size of bacteria in this exciting graphic adventure! Injection into the president's bloodstream, your mission is to destroy the deadly amoeba infection ravaging his body! ★ **LOGIC** — This popular game is based on determining but utilizes tactics that make it more exciting and challenging — has 2 levels of play to make it fun for everyone. ★ **SUBMARINER** — Shoot torpedoes at the enemy ships to get points. Fast action graphics, arcade type game is exciting and fun for everybody!

★ PACKAGE SIX ★

20 HOME FINANCIAL PROGRAMS — Figures amortization, annuities, depreciation rates, interest tables, compound interest on savings and much, much more. These programs will get used again and again. A must for the conscientious, inflation minded person.

★ PACKAGE SEVEN ★

BACKGAMMON 5.0 — 2 different skill levels makes this game a challenge to beginners or advanced players. (Not recommended for beginners!). Looks for best possible move to beat you! FANTASTIC GRAPHICS. Plays doubles and even international rules. ★ **SPEED READING** — Increases your reading speed. Also checks for comprehension of material. Great for teenagers and adults to improve reading skills. ★ **PT 109** — Drop depth charges on moving subs. Learn depths get higher points in this fast action graphics game. ★ **YAHTZEE** — Play Yahtzee with the computer. This popular game is even more fun and challenging against a TRS-80! ★ **WALL STREET** — Can you turn your $50,000 into a million dollars? That's the object of this great game. Simulates an actual stock market!

AVAILABLE AT RETAIL STORES ANYWHERE

INSTRUCTION BOOK WITH EACH PKG.

ONLY 12.95 EACH!!!!

ALL PROGRAMS GUARANTEED TO LOAD

CASSETTE PACKAGES REQUIRE 16K LEVEL II

PACKAGES ON DISKETTE (32K) $5.00 EXTRA

Send check, Money Order or Bank Card #

TO: SIMUTEK, P.O. BOX K 35298
TUCSON, ARIZONA 85740
(602) 882-3948

✓19

PHONE ORDERS WELCOME!

PLEASE ADD $2.50 POSTAGE & HANDLING PER ORDER

3 OR MORE PACKAGES GET 10% DISCOUNT

```
                01530 ;AND WILL WATCH CCNT. IF CCNT IS ZERO THEN
                01540 ;SCAN WILL READ A RECORD. IF EOF IS FOUND OR
                01550 ;ANY READ ERROR IS ENCOUNTERED SCAN WILL
                01560 ;DISCONNECT ITSELF AND THE 25 MS HANDLER
                01570 ;
FDB9 F5         01580 SCAN      PUSH    AF
FDBA E5         01590           PUSH    HL
FDBB 2A3FFE     01600           LD      HL,(CCNT)
FDBE 7D         01610           LD      A,L
FDBF B7         01620           OR      A
FDC0 2004       01630           JR      NZ,EXIT
FDC2 7C         01640           LD      A,H
FDC3 B7         01650           OR      A
FDC4 2B05       01660           JR      Z,RRCD          ;YES READ RECORD
                01670 ;NOPE - RETURN TO KEYBOARD
FDC6 E1         01680 EXIT      POP     HL
FDC7 F1         01690           POP     AF
FDCB C30000     01700           JP      0               ;DUMMY JUMP
FDC9            01710 KEY       EQU     $-2             ;BACK UP 2
FDCB C5         01720 RRCD      PUSH    BC
FDCC D5         01730           PUSH    DE
FDCD DDE5       01740           PUSH    IX
FDCF FDE5       01750           PUSH    IY
FDD1 111AFE     01760           LD      DE,INBFR        ;DCB
FDD4 CD3644     01790           CALL    READ            ;READ RECORD
FDD7 2B17       01800           JR      Z,OKR           ;READ OK SET COUND
                01810 ;NOT OK KILL EVERYTHING
FDD9 F3         01820 CLOS      DI                      ;STOP INTS.
FDDA 2A5FFE     01830           LD      HL,(SAV25)      ;OLD ADDRESS
FDDD 221045     01840           LD      (MS25),HL       ;PUT BACK
FDE0 2AC9FD     01850           LD      HL,(KEY)        ;OLD KBD
FDE3 221640     01860           LD      (KBDO),HL       ;PUT BACK
                01870 ;NOW POP REGISTERS AND RESTORE STACK
FDE6 FDE1       01880 POP       POP     IY
FDE8 DDE1       01890           POP     IX
FDEA D1         01900           POP     DE
FDEB C1         01910           POP     BC
FDEC FB         01920           EI
FDED C3C6FD     01930           JP      EXIT
                01940 ;READ IT OK SET UP CCNT THEN GET OUT
FDF0 2161FE     01950 OKR       LD      HL,BUFFER
FDF3 223DFE     01960           LD      (ADDR),HL
FDF6 2A17FE     01970           LD      HL,(SECTOR)     ;GET SECTORS
FDF9 7D         01980           LD      A,L             ;TEST
FDFA FE00       01990           CP      0               ;ZERO?
FDFC 200D       02000           JR      NZ,DECIT        ;NOPE DEC IT AND STORE
FDFE 7C         02010           LD      A,H
FDFF FE00       02020           CP      0               ;HI =ZERO?
FE01 2008       02030           JR      NZ,DECIT        ;NOPE
                02040 ;SECTOR COUNT=0, USE EOF BYTE COUNT NOT 256
FE03 3A19FE     02050           LD      A,(BCNT)
FE06 6F         02051           LD      L,A
FE07 2600       02060           LD      H,0
FE09 1807       02070           JR      SCNT
FE0B 2B         02080 DECIT     DEC     HL
FE0C 2217FE     02090           LD      (SECTOR),HL
FE0F 210001     02100           LD      HL,256
FE12 223FFE     02110 SCNT      LD      (CCNT),HL
FE15 1BCF       02120           JR      POP             ;RESTORE AND GET OUT
                02130 ;
FE17 0000       02140 SECTOR    DEFW    0
FE19 00         02150 BCNT      DEFB    0
FE1A 20         02160 INBFR     DEFM    ' '
FE26            02170 SEC       EQU     INBFR+12
FE22            02180 BX        EQU     INBFR+8
FE3D 61FE       02190 ADDR      DEFW    BUFFER
FE3F 0000       02200 CCNT      DEFW    0
FE41 44         02210 MSG1      DEFM    'DSPOOL FILESPEC?'
FE51 03         02220           DEFB    3
FE52 44         02230 ERM       DEFM    'DSPOOL ERROR'
FE5E 03         02240           DEFB    3
FE5F 0000       02250 SAV25     DEFW    0
FE61 00         02260 BUFFER    DEFB    0
FD00            02270           END     SETUP
00000 TOTAL ERRORS
25999  TEXT AREA BYTES LEFT


ADDR    FE3D 02190    01330 01390 01960
ALIV    3FFF 00320    01180 01200
BCNT    FE19 02150    00B90 02050
BDTABL  FD72 01030    00440
BUFFER  FE61 02260    00710 01950 02190
BX      FE22 02180    00BB0
CCNT    FE3F 02200    01240 01320 01470 01490 01600 02110
CLOS    FDD9 01820
CNTREG  00EA 00290    00410 01210
CONT    FDAA 01400    01230 01300 01500
CR      FDAD 01440    01370
DECIT   FE0B 020B0    02000 02030
```

low at least 51 unused bytes at the top of memory. Remember the end of the program is not the end of the memory it uses. Both SPOOL and DSPOOL use 256 bytes of memory starting at BUFFER. If BUFFER is at FE69H the program uses memory up to FF69H.

It is also necessary to change the program ORG if you have less than 48K of memory or if a program is already using the top of your memory.

Another useful modification replaces the 32 blanks in INBFR (line 2160 in DSPOOL, line 1210 in SPOOL) with a file name. For example:INBFR DEFW 'PRINTFIL/LST '. (Be sure to include enough spaces after the file name and before the last quote mark to make a total of 32 characters.)

Then delete the code that requests the file specification (lines 500 through 680 in DSPOOL, lines 300 through 450 in SPOOL). The system then uses 'PRINTFIL/LST' as the SPOOL, DSPOOL file and you don't need to answer the filespec question.

Operation

Operating the SPOOL-DSPOOL system is very easy. Assemble the programs and create the disk files using NEWDOS EDTASM, the Radio Shack EDTASM and TAPEDISK or any other assembler. I use SPOOL/CMD as the file name for the spooler and DSPOOL/CMD for the despooler.

To use the system you need only type SPOOL when you want the spooling to begin and answer the FILESPEC? question with the name of the file that is to hold the printer output. If you want to spool BASIC output, you must run SPOOL before you go to BASIC, unless you have NEWDOS.

With NEWDOS you can run the SPOOL-DSPOOL system from BASIC with the CMD"XXX" command. When all of your printer output is spooled return to DOS and type "CLOSE" (or type CMD"CLOSE" from NEWDOS BASIC). When you are ready to print the file type "DSPOOL" and answer the FILESPEC? question with the

Wayne Green

Okay, now you've had a chance to see what I have in mind for you with 80 MICROCOMPUTING. Oh, I admit that we're just getting started and that the magazine will be improving a lot as we go along. We have some interesting ideas in the works for you.

With the TRS-80 (or 90 . . . etc.) being the most popular microcomputer in the entire world, you are going to benefit from this in many ways. The more computers there are out there of one kind . . . the more good programs you are going to have for this system. I hope that is obvious. You may be sure that 80 MICROCOMPUTING will be packed with the shorter programs and reviews of the larger ones. You can waste an awful lot of money on stuff that looks great in the ads, but fizzles out when you try to use it. You need our reviews.

The wealth of programs will also mean that there will be much better programs for the TRS-80 than any other system. Put yourself in the seat of a computer programmer and you'll understand this. If you are going to spend several months developing a comprehensive program, and it takes all of that to write and debug a big program, would you write it for a system which has sold one hundred units or one which has sold over 300,000 systems? The answer is obvious . . . and this is why we are already seeing programs coming out for the "80" which are far better than anything for any other system on the market. This is tough for other systems . . . the law of the computer jungle.

Between our connections with Instant Software, the largest publisher of microcomputer programs in the world, and Kilobaud MICROCOMPUTING, you know that 80 MICROCOMPUTING is going to be your most important link with software for the TRS-80.

With Instant Software being sold and promoted in every country in the world where the TRS-80 is being sold, our input of programs is also the best in the world. We get programs submitted from everywhere . . . often from 50 to 100 a week! You'll get the cream of the crop either published or reviewed in "80".

HARDWARE TOO

The same law of the computer jungle holds for hardware. Would you, as a manufacturer, market an accessory for a system which has sold 100 units or would you go first for the one which has sold hundreds of thousands. It is, as with software, self-evident why the great bulk of the hardware accessories for computers are for the TRS-80 these days.

80 MICROCOMPUTING has the advantage of the use of the largest and most complete microcomputer lab in the world . . . the one developed for Instant Software and Kilobaud MICROCOMPUTING. This means that most new pieces of equipment are tested and in use by our staff . . . and this means that we can tell you what we think is outstanding . . . and where we find ripoffs. This lab is important to you.

SUBSCRIBE

If you are not already a subscriber to 80 MICROCOMPUTING, please get signed up right now. The yearly rates are $15, and that is a bargain. Just one single program or use to you can be worth much more than that. One review of an accessory could save you many times that much investment. I would appreciate it if you would appoint yourself a committee of one to get more subscribers for the magazine. You will benefit even more than we do here at the magazine . . . because the more readers we have, the more ads we will be able to attract . . . and the more ads, the more pages of articles you will get every month.

The "80" market can, I think, support a couple of hundred pages of ads . . . and that would mean a magazine of nearly 500 pages a month. That should hold you. You may not have time left to use your computer.

ENCYCLOPEDIA

If you've read Kilobaud MICROCOMPUTING, you know that I try hard not to duplicate published material. My concept is that every reader should save every issue (we sell inexpensive boxes for this so they can sit on your library shelf) and treat the magazine as a continuing encyclopedia of computing. I make sure that much of the material in each issue is written in simple language so it will be understandable by even the rawest newcomer to computers. Oh, I have articles for the more advanced users too, so you'll have something to look back over later and see as your understanding of your system grows.

Try to think of 80 MICROCOMPUTING as more of a large club newsletter than an ivory tower high-level publication. I'll leave the pomp to other publishers . . . the ones with the well-deserved inferiority complexes who cater to their inadequacies by publishing esoteric baloney. This magazine is written by the readers and edited by people whose aim is to help you enjoy your TRS-80.

SAVE

With each issue costing $2 at your computer store, that's $24 a year. For $15 a year you can subscribe . . . at least for now. As the magazine expands, please do not be surprised if the cover price increases, along with the subscription price. I started 73 Magazine for radio amateurs twenty years ago with a cover price of 37¢ (two for 73¢) and it is up to $2.50 a copy now (and it is the largest of the ham magazines).

For you bargain hunters . . . and those who find that one year goes by all too rapidly, the three year rate for "80" is $36. This, too, will be going up . . . reflecting the inflation, paper increases, postage increases, and a short vacation for me in Hong Kong next year. Someone has to pay for that.

If coupon below has been used, please fill out subscription form on the reader service card in the back of the magazine.

YES! Sign me on as a subscriber to 80 Microcomputing for only $15 a year!

Card # _____ Exp. _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

☐ 12 issues – $15
☐ 36 issues – $36
☐ Please bill me
☐ Payment Enclosed
☐ Master Charge
☐ VISA
☐ American Express

Subscription begins with next published issue.
Back issues, while available, are $3 each.
Canada: $15 per year US funds.
All other foreign subscriptions $20, one year only, US currency only.

80 microcomputing™

PO Box 981 • Farmingdale NY 11737

30388

same filespec used to spool the output. When the system returns to DOS you may run another job, as long as you follow the rules.

While DSPOOL is running, the character in the lower right corner of the TRS-80 video display will flash. This indicates that DSPOOL is running. If you do not like this feature delete lines 1180 through 1200 in DSPODL.

Summary

The source code given in the listings is for the NEWDOS Editor-Assembler. You can easily change the code for any other assembler. Don't forget the rules given above. Always close your spool file when you are finished and be sure to protect the memory used by these programs when in BASIC.

Don't attempt to use CLOSE to close the read file after you run DSPOOL. It's not necessary and won't work.

If you have two disk drives you can use one entire diskette to spool printer output. If you have only one drive, your spooling is limited, but you should be able to accumulate several pages of output before you must DSPOOL. Either way SPOOL-DSPOOL should improve your TRS-80 throughput. ■

```
DISP    4467 00210    00520 00770
ODS     402D 00280    00780 01010
DTAREG  00E8 00300    01350
ERM     FE52 02230    00760
EXIT    FDC6 01680    01630 01930
INBFR   FE1A 02160    00530 00720 01780 02170 02180
INPUT   0040 00220    00550
INTHDL  FD7C 01160    01150
KBDD    4016 00260    00930 00960 01860
KEY     FDC9 01710    00940 01850
MS25    4510 00250    00810 00840 01840
MSG1    FE41 02210    00510
OK      FD49 00800    00750
OKK     FDF0 01950    01800
OPEN    4424 00230    00740
OTPT    FD98 01310    01270
PINT    FD7A 01150    00830
POP     FDE6 01880    02120
READ    4436 00240    01790
RESURT  00E8 00330    00370
RRCD    FDCB 01720    01660
SAV25   FE5F 02250    00820 01830
SCAN    FDB9 01580    00950
SCNT    FE12 02110    02070
SEC     FE26 02170    00860
SECTDR  FE17 02140    00870 01970 02090
SETUP   FD00 00370    00580 02270
SWITCH  00E9 00340    00380 00420 00490
XOR     FDAF 01450
```

```
01210        LD      A,(37E8H)
01215        AND     0F0H
01220        CP      30H
01225        JR      NZ,CONT
```

Program Listing 3.

Artificial intelligence in a 4K Level I? Read on.

4K Intelligence

William M. Lopez
69 East I Street
Chula Vista CA 92010

In the majority of computer games where the player pits his skill against the computer, a random number generator is used to select the move by the computer. Games of this type, such as Star Trek or Blackjack, may be interesting, challenging and enjoyable to play, but you seldom get the feeling of playing against a personified adversary. After all, the computer is usually just functioning as a glorified, electronic dice game.

Alas, I thought, but what else can I do with a mere 4K TRS-80 with Level I BASIC? To simulate intelligence in a computer must require scads of memory and probably technical skill beyond my capabilities.

Or so I thought until I ran across the article "Hexpawn—a Beginning Project in Artificial Intelligence" by Robert R. Wier in *The Bast of Byte*, Vol. 1, p. 309. Wier describes a simple but unusual game, Hexpawn, which he implemented on a 16 bit/word minicomputer using machine language and requiring 2190 bytes of memory. Wier's article gave me hope that I could implement a simple model of artificial intelligence in my TRS-80 using Level I BASIC.

The game was originally described in Martin Gardner's "Mathematical Games" column in the March 1962 issue of *Scientific American*. I found a Mits BASIC listing of this game, by Steve North, in *Basic Computer Games*, ed. David H. Ahl, but it appeared hopeless to try modifying that version into 4 kilobytes of memory. Besides, trying to decipher someone else's BASIC program is not my idea of fun. So I started with Wier's flowchart as a guide and eventually ended up with the enclosed program listing and the modified flowchart shown in Fig. 1.

The Game

The game is played with chess pawns on a 3 by 3 board as shown in Photo 1. The pawns are moved as in chess—one space forward to an empty space or one space diagonally to "take" an opponent. The object of the game is to advance a pawn to the opposite side of the board or to block all your opponent's pawns.

The unusual aspect of this game is the way the computer plays. At first it is ridiculously easy to defeat the computer, but after playing a few games, it becomes apparent that the computer is "learning" to play better and better, and soon it becomes unbeatable.

The key to this behavior is that the program is self-modifying (a necessary condition for artificial intelligence). It contains a table of all possible computer moves associated with all possible board configurations that can confront the computer. Whenever it loses a game, the computer eliminates the move that caused the loss from its repertoire of moves.

The Program

Table 1 contains all the required board configurations (models) that can confront the computer. Wier's equivalent table contains 33 models, but I found the need for six more.

In the program, the status of the board configuration is represented by the variables H and C, which I call model designators. The board positions, numbered 0 through 8, can be thought of as represent-

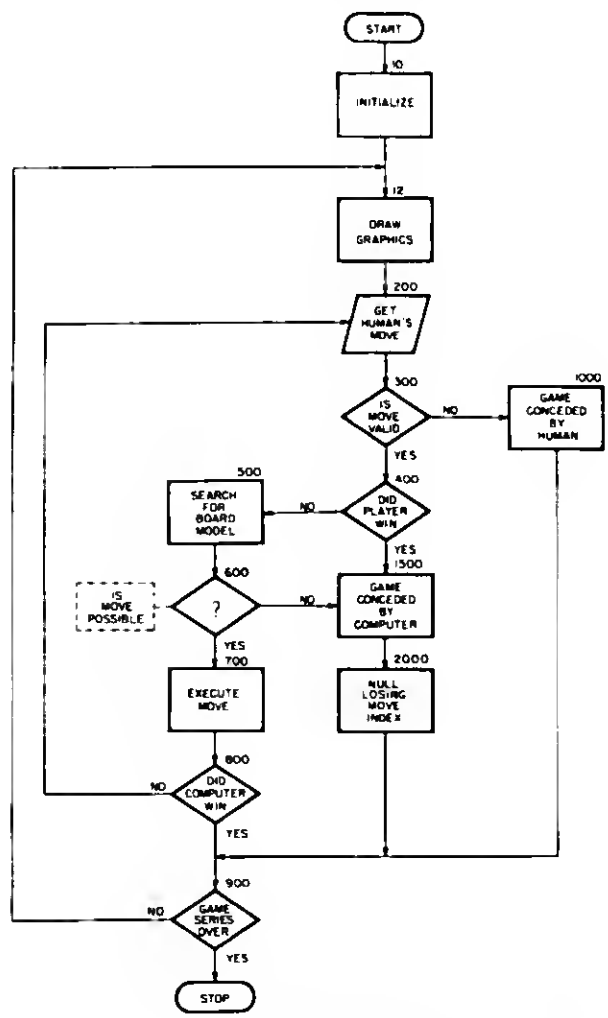


Figure 1. Program flowchart.

Reader Service—see page 147

	Board Positions									Model Designators		Possible Moves
Model #	0	1	2	3	4	5	6	7	8	H_{10}	C_{10}	A(I)
1	C	C	C	H				H	H	392	7	4,7,3
2	C	C	C			H	H	H		224	7	1,4,5
3	C	C	C		H		H		H	336	7	1,2,0
4	C		C	C	H				H	272	13	2,6,8
5		C	C	H	C				H	264	22	3,7,11
6	C		C	H	H			H		152	5	2,8,7
7	C	C		H		H			H	296	3	3,4,5
8		C	C		C	H	H			96	22	5,10,11
9		C	C	C	H	H	H			112	14	5,8,0
10	C		C	C		H		H		160	13	8,9,0
11	C	C		H	H	C			H	280	35	2,3,0
12		C	C	H		H	H			104	6	3,4,5
13		C	C		H			H		144	8	6,7,0
14		C	C		H		H			80	6	6,7,0
15	C		C	H				H		136	5	7,0,0
16			C	C	C	H				32	28	8,11,0
17	C			H	H	H				56	1	2,0,0
18		C		C	H	H				48	10	8,5,0
19		C		H	H	C				24	34	3,14,0
20	C			C	C	H				32	25	8,11,0
21	C		C	H		H				40	5	15,0,0
22	C			H	C			H		136	17	15,0,0
23			C	H	C	C				6	52	11,14,0
24			C	C	H					16	12	6,7,8
25		C		H	C					8	18	3,11,0
26		C		C	H					16	10	5,11,0
27	C		C	H						8	5	2,8,0
28			C		H	C				16	36	8,14,0
29	C			H	H					24	1	2,0,0
30	C		C		H	H		H		176	5	1,2,8
31		C			H					18	2	15,0,0
32	C		C	H	C	H		H		168	21	15,0,0
33			C	H	H	H				56	4	6,0,0
34		C	C		H				H	272	6	6,7,0
35	C		C	H					H	264	5	7,0,0
36	C		C			H	H			96	5	1,0,0
37	C		C		H	C	H			80	37	1,2,14
38	C		C	H		C		H		136	37	14,0,0
39		C			C	H				32	18	11,5,0

Key: C = Computer's pawn occupies square
H = Human's pawn occupies square.

Table 1. All possible board configurations.

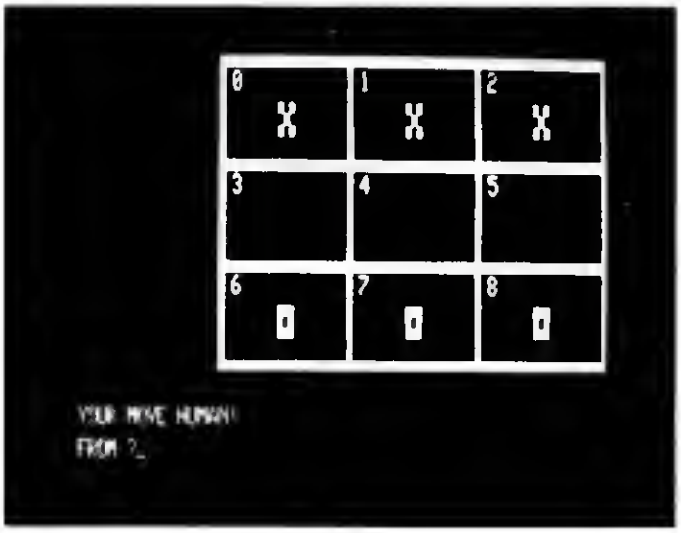


Photo 1. The initial game board configuration. (Photos by Manuel Cavada)

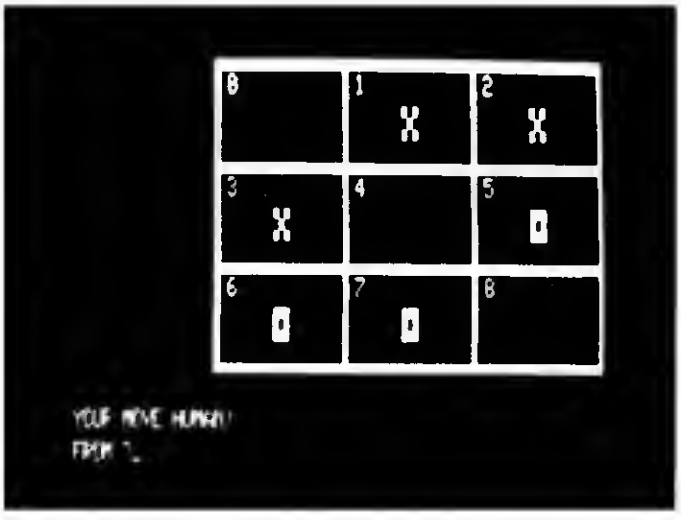


Photo 2. The board configuration after an 8 to 5 opening by the human player and a 0 to 3 response by the computer.

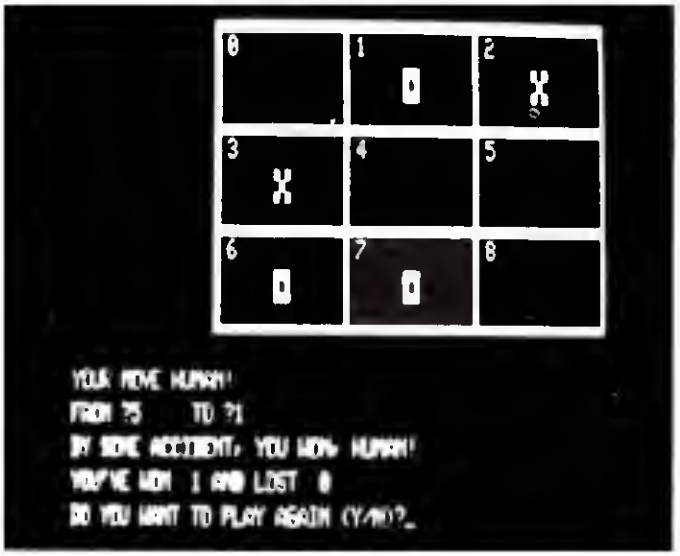


Photo 3. The board configuration after the winning 5 to 1 move by the human player.

ing binary digits with a value of 1 if occupied by a pawn and a value of 0 otherwise. The decimal values of the two binary numbers established by the computer's pawns and by the human player's pawns are stored in the model designators C and H.

At the start of the game, the initial pawn positions shown in Photo 1 establish the values of C = 7 and H = 448. A human move from position 8 to 5 results in a new value of H (program line 410) by the algorithm H = H + 2^5 − 2^8 or H = 224. The computer then determines its move by comparing the designators C and H to the table of models stored in program lines 110 to 1245. It finds a match with model 2 (see Table 1), which yields the first move index A(I) = 1.

From Table 2 we see that this results in the computer move 0 to 3, as shown in Photo 2. If the human player then chooses to move 5 to 1, which results in a win as shown in Photo 3, the program will null the move index "1" for model 2, which eliminates the losing move 0 to 3. Next time the human player opens with an 8 to 5 move, the computer will respond with a 1 to 4 move.

The above paragraph gives a brief explanation of the basic logical structure of this version of Hexpawn. Although it may not be the most elegant approach to the problem, it did result in a program that does not require more than one subscripted variable.

The limitation of Radio Shack's Level I BASIC of allowing only one subscripted variable was somewhat com-

Move Index	Square to	Square	Comments
1	0	3	
2	0	4	
3	1	3	
4	1	4	
5	1	5	
6	2	4	
7	2	5	
8	3	6	Computer wins!
9	3	7	"
10	4	6	"
11	4	7	"
12	4	8	"
13	5	7	"
14	5	8	"
15	-	-	Computer blocked

Table 2. List of computer's moves.

pensated by the ability to abbreviate the BASIC statements. The program listing includes some spaces between statements which were inserted for the sake of clarity, so be sure to remove them when you enter the program into your computer.

If, after entering the program, the "P.M." commend does not indicate a free memory space of 472 or more bytes, the program will stop at line 10 with an error message of "SORRY." Go back and remove more spaces or remove line 1. Since the Radio Shack Level I BASIC does not contain a DIM statement, it apparently cannot determine the size of the array A(I) and it indicates a free memory space.

Playing Hints

A beginning player can usually defeat the computer about ten games before it becomes unbeatable. It becomes a challenge to try to extend your number of wins beyond ten. My record is 18 wins. See if you can match or exceed that.

A word of caution: If you enter your move on the keyboerd before the computer displays the prompt "YOUR MOVE HUMAN!," you may not be able to recover from the resulting mix-up. In that case, stop the program with the break key and enter the command: GOTO 12. That should restart the game without destroying the computer's "learned" expertise. ■

Line Numbers	Purpose
5-9	List of move indices.
15-50	Draw game board on CRT screen.
110-124	List of model designator values.
200-210	Get human move.
305-370	Test for a valid human move. Update C if computer pawn was captured.
402-410	Update graphic display. Check for human win. Update H.
500-502	Search for model matching game board.
505-508	Search for nonzero move index.
600-605	Obtain computer move positions.
700-704	Update graphic display. Update C. Update H if human's pawn was captured.
800-810	Test for computer win.
815-840	Test for blockage of human's pawns.
5000-5010	Subroutine for drawing an X.
5100-5110	Subroutine for drawing an O.
5150-5155	Subroutine for erasing an X or O.
6000-6010	Subroutine for obtaining graphics X,Y coordinates from board position.
6100	Subroutine for testing a board position. K, for occupancy by C or H.
6200-6215	Subroutine for updating H or C after a pawn move.
6400	Subroutine for blanking text from CRT screen.

Table 3. Line descriptions.

Program listing.

```
  1  REM HEXPAWN, TRS-80, 4-K VERSION
  2  M=0:L=0:CLS:P." HEXPAWN":P.
  3  P."DO YOU WANT INSTRUCTIONS (Y/N)";
  4  Y=1:IN.A:IF A=1 GOS.6300
  5  D. 4,7,3,1,4,5,1,2,0,2,6,8,3,7,11,2,6,7,3,4,5,5,10,11,5,6,0
  6  D. 8,9,0,2,3,0,3,4,5,6,7,0,6,7,0,7,0,0,8,11,0,2,0,0,8,5,0
  7  D. 3,14,0,8,11,0,15,0,0,15,0,0,11,14,0,6,7,8,3,11,0,5,11,0
  8  D. 2,8,0,6,14,0,2,0,0,1,2,6,15,0,0,15,0,0,6,0,0,6,7,0,7,0,0
  9  D. 1,0,0,1,2,14,14,0,0,11,5,0
 10  F.I=1TO117:READ A(I):N.I
 12  CLS
 15  F.J=0 TO 2:F.I=0 TO 2:P.A.78+J*192+12*I,3*J+I:N.I:N.J
 20  F.I=0 TO 72
 25  S.(28+I,2):S.(28+I,11):S.(28+I,20):S.(28+I,29)
 30  N.I
 35  F.I=0 TO 27
 40  S.(28,2+I):S.(52,2+I):S.(76,2+I):S.(100,2+I)
 45  N.I
 50  F.I=0 TO 2:X=I:GOS.5000:X=I+6:GOS.5100:N.I
105  C=7:H=448
110  D. 392,7,224,7,336,7,272,13,264,22,152,5,296,3,96,22,112,14
115  D. 160,13,280,35,104,6,144,6,80,6,136,5,32,28,56,1,48,10,24
120  D. 34,32,25,40,5,136,17,8,52,16,12,8,18,16,10,8,5,16,36,24
122  D. 1,176,5,16,2,168,21,56,4
124  D. 272,6,264,5,96,5,80,37,136,37,32,18
200  REST.:P.I=1TO117:READ A:N.I
202  GOS.6400
205  P.A.704,"YOUR MOVE HUMAN!"
210  IN."FROM";J:P.A.780,"TO";:IN.K
305  X=C:GOS.6100
310  IF Z=0 T.360
340  IF(J-K<>2)*(J-K<>4)T.1000
344  X=1:P.I=1TOK:X=2*X:N.I
345  C=C-X:G.400
360  IF J-K<>3T.1000
365  X=H:GOS.6100
370  IF Z<>0 T.1000
400  REM
402  X=J:GOS.5150:X=K:GOS.5150:X=K:GOS.5100
405  IF(K=0)+(K=1)+(K=2)T.1500
410  X=H:GOS.6200:H=X
412  X=0
500  F.I=1 TO 39
501  READ A,B:IF(A=H)*(B=C) X=I
502  N.I
504  IF X=0 T. 3000
505  F.I=0 TO 2
506  Y=3*X-2+I
507  IF A(Y)<>0 T. 515
508  N.I
510  P."I CONCEDE THIS GAME, HUMAN!":G.2000
515  Z=A(Y)
600  IF Z=15 T.1500
601  N=Y
602  D. 0,3,0,4,1,3,1,4,1,5,2,4,2,5,3,6,3,7,4,6,4,7,4,8,5,7,5,8
605  P.I=1 TO Z:READ J,X:N.I
700  X=J:GOS.5150:X=K:GOS.5150:X=K:GOS.5000
701  X=C:GOS.6200:C=X
702  IF(K-J<>2)*(K-J<>4)T.800
704  X=1:P.I=1 TO K:X=2*X:N.I:H=H-X
800  IF H=0 T.890
810  IF Z>7 T.890
815  F.I=3 TO 8
820  K=I:X=H:GOS.6100
825  IF Z=0 T.840
830  K=I-3:X=C:GOS.6100
835  IF Z=0 T.200
840  N.I
890  GOS.6400:P.A.704,"I WIN, HUMAN!"
895  L=L+1
900  P."YOU'VE WON ";W;"AND LOST ";L
907  P."DO YOU WANT TO PLAY AGAIN (Y/N)";
910  Y=1:IN.A:IF A=1 T.12
920  END
1000  L=L+1
1005  P.A.832,"INVALID MOVE DUMMY, I WIN (CHUCKLE)"
1010  G.900
1500  P."BY SOME ACCIDENT, YOU'VE WON, HUMAN!"
2000  A(M)=0:M=M+1:G.900
3000  P."ERROR! C,H=";C;H:STOP
5000  GOS.6000
5010  P.U=0 TO 2:S.(X+U,Y+U):S.(X+U,Y+2-U):N.U:RET.
5100  GOS.6000
5102  F.U=0 TO 2:F.V=0 TO 2
5105  S.(X+U,Y+V)
5110  N.V:N.U:R.(X+1,Y+1):RET.
5150  GOS.6000
5152  F.U=0 TO 2:F.V=0 TO 2
5155  R.(X+U,Y+V):N.V:N.U:RET.
6000  Y=6+INT(X/3)*9
6010  X=39+24*(X-INT(X/3)*3):RET.
6100  F.U=0 TO K:Y=INT(X/2):Z=X/2-Y:X=Y:N.U:RET.
6200  A=1:B=1:IF J=0 T.6210
6205  F.I=1 TO J:A=2*A:N.I
6210  IF K=0 T.6215
6212  F.I=1 TO K:B=2*B:N.I
6215  X=X-A+B:RET.
6300  P."HEXPAWN IS PLAYED WITH CHESS PAWNS ON A 3 BY 3 BOARD,"
6302  P."THE PAWNS ARE MOVED AS IN CHESS - ONE SPACE FORWARD"
6304  P."TO AN EMPTY SPACE OR ONE SPACE DIAGONALLY TO CAPTURE"
6306  P."AN OPPONENT.  MY PAWNS ARE 'X' AND YOURS ARE 'O'."
6308  P."A WIN OCCURS WHEN YOU REACH THE OPPOSITE SIDE OF THE"
6310  P."BOARD OR WHEN YOU BLOCK ALL YOUR OPPONENT'S PAWNS."
6311  P."AN INVALID MOVE RESULTS IN A LOSS.  TO MAKE A MOVE,"
6312  P."ENTER THE PRESENT PAWN POSITION NUMBER, THEN THE NEW"
6314  P."POSITION NUMBER.  ALL KEYBOARD ENTRIES MUST BE COM-"
6316  P."PLETED BY PRESSING THE ENTER KEY.":P.
6318  IN."SHALL WE CONTINUE (Y/N)";A:RET.
6400  F.I=0 TO 3:P.A.704+64*I," ":N.I:RET.
```

A review of three programming aids fickle authors will find interesting.

Useful Utilities

Charles Leedham
114 East 90th
New York NY 10028

If you've done any programming at all, you'll know this situation. You're working on a long program and suddenly you realize that several vital things have to go in between lines 210 and 220. Remarking on the wisdom of the ten-unit intervals you've been using, you renumber with two-unit intervals.

Then something must go in between 210 and 212. Still okay. You renumber with one-unit intervals. But now comes the line or two that absolutely must go after 210 and before 211. Too bad. You could retype 210 as 205, but what about all those GOTO210's? You could retype everything from 211 on with new line numbers. But that would be time consuming.

Renumber

Or you could use Radio Shack's Renumber, a simple aid for programmers who need to make changes or who like to see a clean succession of ten-interval numbers in their programs.

Renumber is a machine-language tape, available for 4K, 16K, 32K and 48K memories, for $9.95 from your local Radio Shack store. The 16K version loads on the SYSTEM command before your working program loads and is called into action by typing /31820 in answer to the *?.

The program then asks you what line number you want to start with, what the new number of that line should be and what interval you want for the remainder of the renumbered program. You can start with 0 and do the entire program or, as in the example, tell it that the old line number was 210, that the new line number should be 210, and that everything from there on should be renumbered at ten-number intervals.

It's done in the twinkling of an eye. So fast, in fact, that when I first used it and the READY came up on the screen, I thought it hadn't worked. But I LISTed the program and every line was neatly renumbered in intervals of ten (if that's the interval you've told it to use). Every reference to a line number in your program is also changed: If your old line 213 is changed to 230 and you have some GOTO213 commands in other sections, all those references will be changed to GOTO230. This is true of all the line-calling commands, including gosubs, on x goto's, etc.

It is really quite neat (literally) —unless you have put a section of the program in, say, line 1000+ and separated others into nice even-hundred-numbered sections and want to keep them that way. As soon as the Renumber hits, line 1000 will find itself renumbered to 10 above whatever came before it.

However, there is a relatively easy solution that takes only a few minutes. Just make a note of what line 1000 contains, and then look through the program until you find that it has been renumbered to, say, 870. Call up the Renumber again and tell it to make 870 into 1000. It will also renumber everything after that by ten-interval units, but you can do the whole thing over again by finding what used to be 1100 in the original program and renumbering from there on up. Somewhat tedious, but it gets the job done.

Remodel-Proload

If you want a really professional programmer's tool, take a look at Remodel-Proload from RACET computes, 702 Palmdale, Orange, CA 92655. It's available for 16K, 32K or 48K memories, but the price is a fairly stiff $34.95.

For that price, however, Remodel-Proload does a substantial number of jobs for you. It will renumber selectively, so that you need only tell it that you want lines 211 through 219 numbered in ten-unit intervals, 210 through 300. If there is a line 300 already, have the program renumber lines 300 through 400 into 410 through 500.

It will also search the program and change all line references to conform to the new numbering system.

One mildly annoying disadvantage of Remodel is the spece it puts before and after a changed line number. A reference line that read GOTO213 ELSE . . . will now read GOTO 230 ELSE . . ., which is bothersome it you want a tightly-packed program for speedy loading and execution. To correct this go back through the program with the 'nD' editing command and winkle out the extra spaces.

Ussge

Now for the Remodel function. Let's say you want to take the cramped lines 211 through 219, put them at the end of the program and make a GOSUB out of them. Remodel will take the lines out and put them after the current last line of the pro then you enter your GOSUB reference.

Remodel can move sections of your program. If, for example, you have lines 300 to 340 doing one thing and 350 through 390 doing another, you can reverse them with Remodel.

Remodel can achieve this two ways. Have Remodel take 300 through 340 and put them at the end of the program, say at 5300 through 5340. Then switch the numbers on 350-390 to 300-340 and tell Remodel to put 5300-5340 back as 350-390. Or, tell it to renumber 350-390 as 295-299 (assuming you don't have these

numbers in the program). Then, start with 295, make it 300, and renumber at ten-unit intervals up to 350, which becomes 390.

Now comes the real fun: Proload. Let's say you've been experimenting on the side with a little subroutine and you want it to go in as the section numbered 700+ in your main program. You don't want to type the whole thing in with 700 numbers. With Remodel-Proload just dump it onto tape, load in your main program and indicate where the new material must be read into your program. Tell the Proload section what you want and load the subroutine tape. If the space indicated is clear, the two programs will be merged.

Example

I'd been working for some days on a special program and decided to take a break to work on a nice little title-with-graphics. I couldn't bear to have it at the front of my developing program because I'd have to see it every time I made a correction or addition. Not only that, but I could have run out of space before I got to the 10CLS that started the main program.

So I fiddled with the title on the side. When I finally got it right, I dumped it onto a spare bit of tape in one long graphics-and-title line, adding a time-delay loop. When I was reasonably happy with the main program I loaded Proload, put in the program, and inserted the title tape as lines 2 and 4, fitting in neatly before the 10CLS.

Having prudently left some space between 70 and a program block beginning at 100, I told Remodel (it's on the same tape with Proload) to renumber from 2 through 70, starting at 10 at intervals of ten. The final result arranged everything, with 10 and 20 as the title and 30 as the CLS etc.

Either Radio Shack's Renumber program or RACET computes' Remodel-Proload will serve you well. The Remodel is obviously better but is 3½ times the price. I use both routinely, always loading Renumber when I'm starting on a program—it loads in just a few seconds. Later I will put in Remodel-Proload if it appears that major surgery will be needed.

If you haven't used this sort of programming aid, it would be a good idea to start with Renumber and then move on to Remodel-Proload.

One small caution about Remodel. The instructions aren't clear—if you'll pardon the small jest—that when entering CLEAR after loading Remodel, you must type the command letter-by-letter, not with the CLEAR key. If you use the key, you'll get an OM error when loading the program tape. Otherwise the manual is concise and well-written.■

PROBLEMS?

Are you having trouble with advertisers or products?
If such is the case, please write to the firm
giving complete details and send a copy of your letter to
Wayne Green, *80 Microcomputing*,
Peterborough, New Hampshire 03458.
Be polite—no matter how badly you're shafted.

We're looking for names and addresses of all the **TRS-80 newsletters** out there. If you produce or are affiliated with such a newsletter, please drop a line to

Jim Perry, Managing Editor
80 MICROCOMPUTING
Peterborough NH 03458

Thanks.

Need to find an article?
This technique was developed by IBM!

KWIC Index

Leslie E. Sparks
1014 Evergreen Dr.
Durham NC 27712

Where's that article on inventory control? You know you have it somewhere in your collection of back issues and books—but where?

If you're like me you go through this quite often.

The KWIC (key word in context) index described in this article can help you get on top of your information explosion. Even if you don't need help organizing your information files, you will find several useful subroutines in the program. Subroutines for chained lists, shell sorts and binary searches are all used in tha program and discussed in the text.

Description of KWIC Index

The KWIC index was developed by IBM to locate specific titles from lists of books, chapters or articles. For example, I want to locate all the articles on the TRS-80 in my files. I can either search my collection of back issues (which is how I did it before I wrote this program) or consult my KWIC index and find all the articles with TRS-80 in the title. The KWIC index is arranged alphabetically by each word in each title.

Take another example: The

	INDEX WORD	REF#
	A Look at TRS-80 Peripherals	3
A Look	at TRS-80 Peripherals	3
	Graphing with the TRS-80	1
A	Look at TRS-80 Peripherals	3
A Look at TRS-80	Peripherals	3
	Sargon Meets the TRS-80	2
Graphing with	the TRS-80	1
Sargon Meets	the TRS-80	2
Graphing with the	TRS-80	1
Sargon Meets the	TRS-80	2
A Look at	TRS-80 Peripherals	3
Graphing	with the TRS-80	1

	INDEX WORD	REF#
	Graphing with the TRS-80	1
A Look at TRS-80	Peripherals	3
	Sargon Meets the TRS-80	2
Graphing with the	TRS-80	1
Sargon Meets the	TRS-80	2
A Look at	TRS-80 Peripherals	3

Figure 1. Example of KWIC Index.

	INDEX WORD	REF#
	Graphing with the TRS-80	1
A Look at TRS-80	Peripherals	3
	Sargon Meets the TRS-80	2
Graphing with the	TRS-80	1
Sargon Meets the	TRS-80	2
A Look at	TRS-80 Peripherals	3

Figure 2. Revised KWIC Index.

REFERENCES

REF#	Author, Title, Reference
1	Gerald, C. F. Graphing with the TRS-80 *Kilobaud* #29 p 100
2	Bobo, R. H. Sargon Meets the TRS-80 *Kilobaud* #31 p 58
3	Cowan, R. A Look at TRS-80 Peripherals *Kilobaud* #28 p 22

Author listing

REF#	
2	Bobo, R. H. Sargon Meets the TRS-80 *Kilobaud* #31 p 58
3	Cowan, R. A Look at TRS-80 Peripherals *Kilobaud* #28 p 22
1	Gerald, C. F. Graphing with the TRS-80 *Kilobaud* #29 p 100

Figure 3. The complete KWIC Index.

realsoft™ PROGRAM STORE

(202) 337-4691

4200 Wisconsin Ave NW PO Box 9609 Washington D C 20016

"This TRS-80* fascinates my whole family!"

50 unique programs for all ages

A package of fifty different game, quiz, instructional, educational, financial and fun programs are available for any TRS-80. Every member of your family will enjoy the TRS-80 with this set of programs called "The Fascinating Fifty." And it is reasonably priced at $24.95.

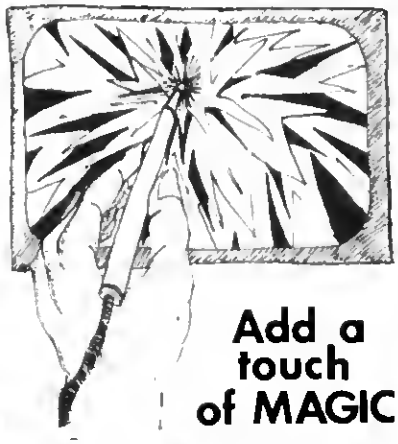
Designed for the whole family, these programs are fascinating. Have fun playing MANCALA or ZOO BREAK, guessing WORLD RECORDS or U.S. PRESIDENTS, learning MOON PHASES or WORD COMPLETION and using FINANCIAL TIMETABLE or METRIC CONVERSION. Other programs include: Rocket, Bingo, Dice, Reverse, World Population, World Area, States, Air Distance Between Cities, World Population by City, Grade Averages, Number Base, Accounts Receivable, Loan Amortization, Interest From Savings, Business Ledger, Calendar, World Clock, Blackboard I, II, III & IV, Maze, Letter Recognition, Numeral Recognition, Right Reading, Word Recognition, How Many, Add On, Take Away, Learn To Spell, Sounds Like, Story Telling, Multiply, Divide, Fractions, State Capitals, Foreign Capitals, Cub Reporter, Spelling - Grade 4, 5 & Jr. High, and Test Your Vocabulary.

If you would like every member of the family to enjoy your TRS-80, today order your copy of "Fascinating Fifty" by Neal Jensen for only $24.95.

Galactic Empire

by Douglas G. Carlston

This is fantastic! "Galactic Empire" opens up a three dimensional universe for exploration and conquest. While travelling through space you command the Imperial Forces of Galactica, your ship's senior officers and its computer. You must plan you construction of additional support craft and their arrival at nearby planets. Increase your strength by taxing and enlisting additional troops from the varied and unique planets you conquer. Besides maneuvering in a three-dimensional universe which changes from game to game, your use of time to travel between planets is very important. Good graphics and the sophisticated strategy make this game much different from any "Star Trek" or "Star Wars." Explore the galaxy for $14.95. You'll love it!



Add a touch of MAGIC with a Light Pen.

With only a few lines of BASIC, any program can accept inputs by touching the light pen on the video screen. Hundreds of applications! For only $19.95 you receive the light pen, instructions on using it in your programs, several sample programs, and a cassette of three light pen programs. This is the standard pen which several software companies support.

DUNJONQUEST

Datestones of Ryn

from Automated Simulations

In this microquest you are Brian Hammerhand on a mission into the Haunted Mountains and Underground lair of Rex the Reaver and his band of thieves. A perfect introduction into the Dunjonquest series which started with "The Temple of Apshai." 16k $14.95.

Also available "Temple of Apshai." $24.95.

Invader

from Level IV

Machine language arcade game with sound. $14.95

Adventure

by Gordon Letwin from Microsoft

The complete Adventure is the ultimate fantasy game which allows you to explore the "Colossal Cave." This is the same as originally written for the DEC PDP-10. It fills an entire disk and requires 32k. For just $29.95

Video Speed Reading

from Instant Software

Using the scientific principle behind the tachistoscope, a mechanical device used to flash characters or words on a screen, this three-part package will train your eyes and mind to quickly recognize numbers, letters, words and phrases. $7.95

Personal Inventory

from Southern States Systems

Designed to allow you to develop an inventory of your personal possessions for insurance purposes, you record for each item its description, serial number and value. Permits sorting, modifying and other record manipulations with storage on cassette tape. $12.50.

TRS 80 Disk & Other Mysteries

by H.C. Pennington

We don't usually list books, but this one is so unique that we thought you would want to know about it. There are over 100 pages about how DOS works, how a disk is organized, and how to recover from errors. This is THE technical backup for NEWDOS+ with great illustrations. $19.95.

Disk*Mod

by Roy Soltoff from Misosys

This machine language program modifies your copy of the Radio Shack Editor/Assembler for use with your minidisk and any disk operating system. You can save and load both text source and assembled object files. Unlike the NEWDOS+ version you can read the directory and the space used and available while in the EDTASM. You can also kill files. It is a complete disk modification for one or more drives.

Other capabilities are also added which are not found on NEWDOS+. The block move command relocates a section of text to any other area. The global change command permits, for example, changing a label throughout the text. The pagination feature provides hardcopy on 8 1/2 by 11 pages on either single sheets or continous paper. In addition, high memory can be reserved, like in BASIC, for machine language routines like printer drivers. You can also display the amount of memory remaining.

The <CLEAR> key is functional, the symbol table is sorted alphanumerically and output 5-across, the scroll up/down allows 15 lines on the screen, and the 'DEFM' assembly is improved. Lower case input is now permitted and you can branch to any address. Plus, it also corrects the errors in the Radio Shack tape version.

Save your time and make full use of your disk system by upgrading your Editor/Assembler today. $19.95

System Doctor

from The Bottom Shelf

Assure yourself that your TRS-80 is working properly. The causes of hardware problems are often hard to detect, but you can now do a thorough diagnostic check of your entire computer. It checks the ROM, the RAM, disk drives, video memory and display and cassette recorder. Also provided is a special test tape and a 12-hour test with a recording of results to find those occasional errors. $28.50.

On disk $38.50

THE PROGRAM STORE
4200 Wisconsin Ave NW
PO Box 9609 Dept K 6
Washington, D.C. 20016

YES please send me these TRS-80 programs:

title	price
postage:	$ 1.00
total:	

name: _____

address: _____

city, state & code _____

☐ Check payable to The Program Store
☐ MASTERCHARGE mc bank code:
☐ VISA exp date:
card number:
signature:

```
KWIC program

Input all authors, titles and references
assign reference number to each article
sort articles by author
Output numerical list of articles
Output list of articles sorted by author
Input list of ignored words
sort ignored words
Do for each article title
    do for each word in title
        if word is not on list of ignored words
            then add to key word list
            endif
        enddo
    enddo
sort list of key words
Print title corresponding to each entry
endprogram KWIC
```

Figure 4. KWIC Index algorithm.

```
SUBROUTINE INITIALIZATION
initialize list head and start of free list
    M2 = 2 (M2 is pointer to available location)
    PRT(1) = 0
    set pointers for free list
    do for I = 2 to N - 1
        PTR(I) = I + 1
        enddo
    set null pointer at end of list
    PRT(N) = 0
return
endsubroutine INITIALIZE
```

Figure 5. Algorithm for initialization of chained list.

```
SUBROUTINE INSERT
  Program to insert the name NAME in the ordered list contained in arrays DATA and
PTR. Head of data list is in PTR(1), head of free list is M2
    if M2 = 0
        then print "NO FREE SPACE" return
        else
            I = 1
            search list for insertion point
            do while PTR(I)≠0 and DATA(PTR(I))<NAME
                I = PTR(I)
                enddo
                I now contains entry of last element in list less than NAME.
                Allocate space from free list for new entry and insert it
                following entry I by setting pointers
                J = M2
                M2 = PTR(J)
                PTR(J) = PTR(I)
                DATA(J) = NAME
                PTR(I) = J
            endif
        return
    endsubroutine INSERT
```

Figure 6. Subroutine for inserting data in order list.

following articles appeared in recent issues of *Kilobaud Microcomputing:* "Graphing with the TRS-80," "Sargon Meets the TRS-80," "A Look at TRS-80 Peripherals." Each article can appear in the KWIC index once for each word in the title. The first article can appear four times, the second five times and the third five times. A KWIC index for these three articles is given in Figure 1.

Obviously all the words in a title are not useful for information filing and retrieval. Eliminate such words as: A, at, on, by, before constructing your KWIC index. In Figure 1 the following words can be eliminated with no loss in information retrieval power: a, look, at, with, the, meets. Our revised KWIC index is given in Figure 2.

The complete KWIC index consists of three parts: a listing

```
SUBROUTINE PRINT
      Program to print the chained list contained in arrays DATA and PTR
      List head assumed to be in PTR(1)
            I = PRT(1)
            do while I≠0
                  output DATA(I)
                  I = PTR(I)
                  enddo
            return
endsubroutine PRINT
```

Figure 7. Subroutine to print ordered list.

```
SUBROUTINE BUBBLE SORT
      Program to sort array DATA using bubble sort.
      I = N
      K = 1
      The flag K is nonzero on the first pass of the outer loop and whenever switches
      are made on the previous pass
      do while I≥2 and K≠0
            the largest N – I elements are now in order in positions DATA(I + 1) to
            DATA(N). Float the largest of the elements of DATA(I)
            J = 1
            K = 0
            do while J<I – 1
                  if DATA(J)>DATA(J + 1)
                        then
                              switch entries DATA(J) and DATA(J + 1)
                              TEMP = DATA(J)
                              DATA(J) = DATA(J + 1)
                              DATA(J + 1) = TEMP
                              K = 1
                        endif
                  J = J + 1
            enddo
      I = I – 1
enddo
```

Figure 8. Bubble sort.

of the articles in numerical order of reference number (I give all my articles a reference number); a listing of articles in alphabetical order by author; and the KWIC index itself.

The complete KWIC index for the example is shown in Figure 3.

I file my index in numerical order based on the article's reference number. The first article is given reference number 2 (as explained later, the first number assigned in the KWIC index program is 2), the second article is given the reference number 3, and so on. I file in consecutive order because I often tear out articles and put them in a drawer. Filing new articles by number and relocating them is simple. Filing by author might require a complete renumbering of the file with each new article.

Constructing the KWIC index

First enter and store the article titles, authors and references. Next, enter and store the list of words to be ignored. Sort this list alphabetically for fast searching.

Next, take the titles apart word by word in search of key words. Words on the ignored list are discarded. Those remaining are stored and sorted alphabetically. The KWIC index is printed as shown in the examples. Finally, everything is stored on disk or tape for future use.

The algorithm

The algorithm for the KWIC index, a slight modification of the algorithm presented by C. William Gear in *Applications and Algorithms in Computer Science*, is given in Figure 4.

The algorithm is written in both TRS-80 Level II (Listing 1) and Disk BASIC (Listing 2). Both programs assume you have a line printer.

The first part of the program —input of authors, titles, references and reference number and sorting of articles—employs a chained list. This is used to avoid moving large

```
SUBROUTINE BINARY SEARCH
    This subroutine searches the array W1 (list of ignored words) to see if word W is
    on the list. If W is on the list, it is discarded. If W is not on the list it is saved as
    a key word
        I = 0
        Low = 1
        High = N
        do while Low<High and I = 0
            Mid = (Low + High)/2
            if W1(Mid) = W
                then discard W and return
                else if W1(Mid)<W
                    then Low = Mid + 1
                    else High = Mid - 1
                    endif
            endif
        enddo
    if I≠0
            then save W as key word
    return
endsubroutine BINARY SEARCH
```

Figure 9. Binary search.

blocks of data around in memory.

In a chained list a pointer keeps track of author, title, reference and reference number, as the list is sorted. Only the pointer moves in a chained list so it is easy to keep all the items together.

Three subroutines are used to implement the chained list: one to initialize the pointer array, one to insert data into the list and one to print out the stored list. The algorithms for each subroutine are given in Figures 5, 6 and 7. These are taken from Gear.

Entering and sorting the list of ignored words uses a bubble sort. The algorithm for the bubble sort is given in Figure 8.

After data entry the hard part begins. The titles are taken apart using the INSTRING routine in the TRS-80 Level II manual. This routine uses the MID$ and LEN string functions in TRS-80 Level II BASIC to locate the space between words.

Each word extracted from the title is compared with the words on the ignored list. This comparison is accomplished using a binary search, see Figure 9. If the word is not on the list, it is stored for later processing.

Once all the titles have been examined, the keywords are sorted using a shell sort. (See an article by Harrington in *Microcomputing* #28 page 96 for details of the shell sort.) In the shell sort the title identification number stays with the keyword as the sort takes place.

Once the keywords are sorted, the KWIC index can be printed. To do this the keywords must be located. To seve memory, I perform this step using the INSTRING routine. Though the method isn't slow, if you do went faster execution, you can add an array, such as Gear uses, that tracks the key word position.

Once the KWIC index is printed you can save the results on tape or disk. I suggest that you use separate tapes for each list.

Using the program

The KWIC index program is easy to use. Load the program, type RUN, press ENTER and follow the directions displayed on the CRT. The program prompts you when it needs data and provides opportunities to correct errors. Printing data appears simultaneously on the CRT.

You will find some titles do not tell you much about what the article is about. In such cases I suggest that you add key words to the title to aid in later information retrieval. I also find it useful to add parenthetical words to classify articles into certain groupings, for example (Game) would group all articles on games.

How many articles can you handle with the program? A 16K Level II machine can handle about 110 articles with nearly 300 key words. A 32K machine can handle much more since all

```
 2            PAPE R L        WHIP FILE WIPEOUTS
KILOBAUD #31 PAGE 39
 3            MYERS F E       DATA FILE CREATION PROGRAM
KILOBAUD #31 PAGE 44
 4            GUPTON J A JR.  COMPUTER CAREERS IN CAROLINA
KILOBAUD #31 PAGE 48
 5            MCCLURE J       PERSONAL FINANCE SYSTEM PART2
KILOBAUD #31 PAGE 50
 6            BOBO R H        SARGON MEETS THE TRS80 (REVIEW)
KILOBAUD #31 PAGE 58
 7            BROONER E       PROJECTING FUTURE PROFITS
KILOBAUD #31 PAGE 63
 8            SCHWARTZ M      AN INTRODUCTION TO MICROFILMING
KILOBAUD #31 PAGE 122
 9            CHAMBERLAIN B.S.                OSI SUPERBOARD-II  (REVIEW)
KILOBAUD #31 PAGE 66
ALPHABETICAL BY AUTHORS LIST OF REFERENCES ARRANGED AS FOLLOWS
REF#, NUTHOR; TITLE, REFERENCE

 6   BOBO R H.  ; SARGON MEETS THE TRS80 (REVIEW) , KILOBAUD #31 PAGE 58
 7   BROONER E  , PROJECTING FUTURE PROFITS ; KILOBAUD #31 PAGE 63
 9   CHAMBERLAIN B S  , OSI SUPERBOARD-II  (REVIEW) ; KILOBAUD #31 PAGE 66
 4   GUPTON J A JR.  , COMPUTER CAREERS IN CAROLINA , KILOBAUD #31 PAGE 48
 5   MCCLURE J  ; PERSONAL FINANCE SYSTEM PART2 ; KILOBAUD #31 PAGE 50
 3   MYERS F E  , DATA FILE CREATION PROGRAM  ; KILOBAUD #31 PAGE 44
 2   PAPE R L  , WHIP FILE WIPEOUTS , KILOBAUD #31 PAGE 39
 8   SCHWARTZ M  , AN INTRODUCTION TO MICROFILMING , KILOBAUD #31 PAGE 122


LIST OF IGNORED WORDS NOTE WI(1) IS NULL STRING
                1
AN              2
FOR             3
IN              4
MEETS           5
PROGRAM         6
SYSTEM          7
THE             8
TO              9
WHIP            10

                              INDEX                              REF#
                                                                    0
SARGON MEETS THE TRS80        (REVIEW)                             6
     OSI SUPERBOARD-II        (REVIEW)                             9
               COMPUTER       CAREERS IN CAROLINA                  4
   COMPUTER CAREERS IN        CAROLINA                             4
                              COMPUTER CAREERS IN CAROLINA         4
             DATA FILE        CREATION PROGRAM                     3
                              DATA FILE CREATION PROGRAM           3
                  WHIP        FILE WIPEOUTS                        2
                  DATA        FILE CREATION PROGRAM                3
              PERSONAL        FINANCE SYSTEM PART2                 5
            PROJECTING        FUTURE PROFITS                       7
                    AN        INTRODUCTION TO MICROFILMING         8
    AN INTRODUCTION TO        MICROFILMING                         8
                              OSI SUPERBOARD-II (REVIEW)           9
PERSONAL FINANCE SYSTEM       PART2                                5
                              PERSONAL FINANCE SYSTEM PART2        5
     PROJECTING FUTURE        PROFITS                              7
                              PROJECTING FUTURE PROFITS            7
                              SARGON MEETS THE TRS80 (REVIEW)      6
                   OSI        SUPERBOARD-II (REVIEW)               9
      SARGON MEETS THE        TRS80 (REVIEW)                       6
            WHIP FILE         WIPEOUTS                             2
```

Figure 10. Program output.

the additional 16K of memory is available for data storage. If you have a 32K or 48K computer be sure to clear sufficient string space in statement number 90. Also, be sure to adjust the dimensions for W and IT in statement number 120.

Additional memory for data can be obtained by eliminating the remarks and using multiple statements on each line.

Searching for key words is time-consuming, as is shell sort. Because these portions of the program do not require your attention, you can let the program run at night or while you're eating.

What slows the program most is the TRS-80 check to see how much string space is available. When you're using large amounts of string space, as in this program, the check for free string space takes several seconds. Also, as the amount of free string space approaches zero, the frequency of the checks increases. Thus, if you are near the limits of string space, the program may run

several hours. For this reason, it is important that you clear plenty of string space in statement number 90.

An example output for the program is given in Figure 10. ■

Listing 1.

```
10 CLS
20 REM KWIC INDEX PROGRAM  VERSION 3 AUGUST 3 1979
30 REM REF APPLICATIONS AND ALGORITHMS IN COMPUTER SCIENCE
40 REM BY C.W. GEAR SCIENCE RESEARCH ASSOC. 1970
50 REM TRS 80 LEVEL II PROGRAM BY L. E. SPARKS
60 REM VARIABLES A$=AUTHOR,T$=TITLE,R$=REFERENCE,
   W$=KEY WORD,IT=TITLE NUMBER,W1$=WORD TO BE IGNORED
70 REM IN =INDEX NUMBER,M=NUMBER OF TITLES,N=NUMBER OF WORDS TO IGONORE
   NA=NUMBER OF AUTHORS
80 REM CLEAR STRING SPACE
90 CLEAR 6000
100 DEFINT I,J,K,L,M,N:DEFSTR A,R,W,T :REM DEFINE INTEGER AND STRING
   VARIABLES
110 INPUT"HOW MANY TITLES NOTE THAT THIS IS TOTAL NUMBER OF TITLES
   INCLUDING ANY STORED FROM PREVIOUS RUNS";M
120 DIM A(M+1),T(M+1),R(M+1),W(270),W1(110),IN(M+1),IT(270)
130 GOSUB2810  REM INITIALIZE CHAINED LIST
140 IS=1
150 INPUT"ARE YOU ADDING TO A LIST THAT IS STORED ON TAPE";Y$
160 IF Y$<>"Y" THEN 200
170 INPUT"PLACE TAPE ON RECORDER AND PRESS ENTER TO CONTINUE",Z
180 REM READ DATA FROM TAPE
190 INPUT#-1,NA
200 NA=NA-1
210 FOR I=1TONA
220 INPUT#-1,A,T,R
230 PRINT A;" ",T," ",R
240 GOSUB2090  REM INSERT DATA FROM CASSETTE INTO CHAINED LIST
250 NEXT I
260 REM CONTINUE
270 FOR I=1STOM+1
280 L=1
290 CLS
300 FORI=IS TO M
310      INPUT"AUTHOR (END TO STOP)",A
320     IF A$="END" GOTO430
330      INPUT"TITLE",T
340     INPUT"REFERENCE";R
350     CLS:PRINT"AUTHOR ";A
360     PRINT"TITLE ";T
370     PRINT"REFERENCE ";R
380     INPUT"IS THIS CORRECT";Y$
390     IF Y$="Y" GOTO410
400     PRINT"REENTER INCORRECT INFORMATION".GOTO310
410     GOSUB 2090 .REM INSERT DATA IN CHAINED LIST
420 NEXT I
430 NA=NA+I-1
435 GOSUB5000   REM PRINT OUT NUMERICAL LIST OF ARTICLES
440 GOSUB2250   REM PRINT OUT ALPHABETICAL BY AUTHORS LIST
450 REM ENTER WORDS TO BE IGNORED
460 INPUT"ARE YOU ADDING WORDS TO BE IGNORED TO AN EXSTING LIST";Y$
470 IF Y$="N" THEN 550
480 INPUT"PLACE TAPE WITH WORDS TO BE IGNORED ON RECORDER AND PRESS
    ENTER TO CONTINUE",Z
490 INPUT#-1,NI
500 FORI=1TONI
510    INPUT#-1,WI(I)
520 NEXT I
530 N=I-1
540 GOTO 630
550 INPUT"HOW MANY WORDS ARE TO BE IGNORED ";N
560 II=1
570 FOR I=II TO N
580     INPUT"WORD TO BE IGNORED ZZZ TO STOP";WI(I)
590    IF WI(I)="ZZZ" THEN 610
600 NEXT I
610 N=I-1
620 CLS
630 PRINT" THE FOLLOWING WORDS WILL BE IGNORED , INDEX NO"
640 GOSUB 1780
650 NI=N
660 INPUT"DO YOU WANT TO CHANGE ANY OF THESE WORDS";Y$
670 IF Y$<>"Y" THEN GOTO730
680 INPUT"INDEX NO OF WORD TO CHANGE ",I
690 INPUT "NEW WORD ",WI(I)
700 INPUT"ARE THERE ANY MORE WORDS TO CHANGE ",Y$
710 IF Y$="N"GOTO630
720 GOTO 680
730 INPUT"DO YOU WISH TO ADD TO THIS LIST ",Y$
740 IF Y$="N" THEN GOTO 790
750 INPUT"HOW MANY WORDS DO YOU WISH TO ADD",NI
760 II=N
770 N=NN+II
780 GOTO 570
790 REM NOW CONSTRUCT INDEX.  FIRST TAKE TITLE APART TO FIND INDIVIDUAL
   WORDS
800 M2=IN(1)
810     TX=T(M2)
820     TY=" "
830     GOSUB1960
840    IF MI=0 GOTO910
850    W=LEFT$(TX,MI+LEN(TY)-2)
860    GOTO920
870    TX=RIGHT$(TX,LEN(TX)-LEN(W)-1)
890    GOSUB 1960
895    IF MI=0 GOTO 910
900    GOTO 850
910    W=TX
920 REM CHECK AND SEE IF WORD IS ON DELETE LIST
930 GOSUB 2300
940    IF W="" THEN 960
950 IF MI<>0 THEN 870
960 M2=IN(M2)
970 IF M2=0 THEN 990
980 GOTO 810
990 CLS:REM OUTPUT INDEX
1000 LPRINT TAB(47)"  INDEX                                                REF#"
1010 TC=""
1020 GOSUB1500
1030 REM NOW PRINT OUT THE INDEX
1040 FOR I=1 TO L
1050     TX=T(IT(I))
1060     TY=" "
1070     GOSUB1960
1080     IF MI=0 THEN 1140
1090    TW=LEFT$(TX,MI+LEN(TY)-2)
1100    TX=RIGHT$(TX,LEN(TX)-LEN(TW)-1)
1110    IF W(I)= TW THEN 1190
1120    TC=TC+" "+TW
1130    GOTO1070
1140    TW=TX
1150    IF MI=0 GOTO1170
1160     TC=TC+" "+TW
1170    IF TW<>W(I) THEN 1270
1180    TX=""
1190    Z=40-LEN(TC)
1200    IF Z<0 Z=0
1205   TC=STRING$(Z," ")+TC
1210    IF MI=0 THEN TX=""
1220    TX=TW+" "+TX
1230    X=30-LEN(TX) IF X<0 THEN X=0
1240    TX=TX+STRING$(X," ")
1250    LPRINT TC,TX,"         ";IT(I)
1260    TC=""
1270 NEXT I
1280 INPUT"DO YOU WISH TO SAVE EVERYTHING ON TAPE (Y OR N)";Y$
1290 IF Y$="N" THEN 1570
1300 REM SAVE AUTHOR,TITLE,REFERENCE,AND ID NUMBER ON TAPE
1310 CLS
1320 PRINT"PLACE TAPE FOR AUTHOR,TITLE,REFERENCE IN RECORDER"
1330 INPUT"PRESS ENTER TO CONTINUE",Z
1340 PRINT#-1,NA
1350 FORI=2TONA+1
1360 IF A(I)="" THEN 1400
1370    PRINT#-1,A(I),T(I),R(I)
1380   PRINT I;" ";A(I);" ",T(I)," ",R(I)
1390 NEXT I
1400 REM NOW SAVE THE KEY WORDS
1410 INPUT"PLACE KEY WORD TAPE ON RECOREDER AND PRESS ENTER TO
    CONTINUE",Z
1420 PRINT#-1,L
1430 FOR I=2TOL
1440   IF W(I)="" THEN 1480
1450   PRINT#-1,W(I),IT(I)
1460   PRINT W(I),IT(I)
1470 NEXT I
1480 REM NOW STORE WORDS TO BE IGNORED
1490 INPUT"PLACE TAPE FOR WORDS TO BE IGNORED ON RECORDER &
    PRESS ENTER TO CONTINUE",Z
1500 PRINT#-1,NI
1510 FORI=1TONI
1520   IF WI(I)="" THEN 1560
1530   PRINT#-1,WI(I)
1540   PRINT WI(I),
1550 NEXT I
1560 REM COMPLETED WORK
1570 END
1580 REM SHELL SORT OF INDEX WORDS
1590 IM=L-1
1600 IM=INT(IM/2)
```

```
1610 IF IN=0 THEN 1760
1620 J=1
1630 K=L-IN
1640 I=J
1650 IL=I+IN
1660 IF W(I)<W(IL) THEN 1730
1670 TE=W(I):IE=IT(I)
1680 W(I)=W(IL):IT(I)=IT(IL)
1690 W(IL)=TE:IT(IL)=IE
1700 I=I-IN
1710 IF I<1 THEN 1730
1720 GOTO 1650
1730 J=J+1
1740 IF J<=K THEN 1640
1750 GOTO1600
1760 RETURN
1770 STOP
1780 REM SUBROUTINE TO SORT IGNORED WORDS
1790 NI = N
1800 S=0
1810 NI=NI-1
1820 FOR J=1 TO NI
1830    IF WI(J)<=WI(J+1) THEN 1880
1840    WI=WI(J)
1850    WI(J)=WI(J+1)
1860    WI(J+1)=WI
1870    S=1
1880 NEXT J
1890 IF S=1 THEN 1800
1900 REM PRINT LIST
1910 FOR X=1 TO N
1920    PRINT WI(X),X
1930 NEXT X
1940 PRINT
1950 RETURN
1960 REM INSTRING SUBROUTINE REF LEVEL II MANUAL
1970 FOR MI=1 TO LEN(TX)-LEN(TY)+1
1980    IF TY=MID$(TX,MI,LEN(TY)) RETURN
1990 NEXT MI
2000 MI=0:RETURN
2010 REM INITIALIZE CHAINED LIST
2020 M2=2
2030 IN(1)=0
2040 FORI=2 TOM+1
2050    IN(I)=I+1
2060 NEXT I
2070 IN(M+1)=0
2080 RETURN
2090 REM SUBROUTINE TO INSERT DATA IN ORDERED LIST
2100 REM M2=POINTER TO FREE SPACE
2110 IF M2=0 THEN 2240
2120 M1=1
2130      IF IN(M1)=0 THEN 2170
2140      IF A$(IN(M1))>A THEN 2170
2150      M1=IN(M1)
2160 GOTO2130
2170 REM NOW INSERT DATA INTO LIST
2180 J=M2
2190 M2=IN(J)
2200 IN(J)=IN(M1)
2210 A(J)=A:T(J)=T:R(J)=R
2220 IN(M1)=J
2230 RETURN
2240 PRINT"NO FREE SPACE ":RETURN
2250 REM SUBROUTINE TO PRINT ORDERED LIST
2260 LPRINT"ALPHABETICAL BY AUTHORS LIST OF REFERENCES AS FOLLOWS:"
2270 LPRINT"REF#, AUTHOR, TITLE, REFERENCE"
2280 LPRINT" "
2290 M2=IN(1)
2300 REM DO WHILE M2<>0
2310 LPRINT M2," ";
2320    LPRINT A(M2)," , ",
2330   LPRINT T(M2)," , ",
2340    LPRINT R(M2)
2350    M2=IN(M2)
2360   IF M2 = 0 RETURN
2370   GOTO 2300
2380 REM BINARY SORT TO SEE IF WORD IS ON LIST
2390 K=0
2400 IL=1
2410 IH=N
2420  IF IL>IH THEN 2560
2430   IM=INT((IL+IH)/2)
2440   IF WI(IM)=W THEN RETURN
2450   IF WI(IM)>W THEN 2480
2460   IL=IM+1
2470  GOTO 2420
2480   IH=IM-1
2490  GOTO 2420
```


✓ Reader Service—see page 147

```
2500 W(L)=W
2510 IT(L)=M2
2520 L=L+1
2530 RETURN
5000 REM SUBROUTINE TO PRINT NUMERICAL LISTING
5010 FOR I=2TONA+1
5020    LPRINT I,A(I),T(I)
5030    LPRINT R(I)
5040 NEXTI
5050 RETURN
```

Listing 2.

```
10 REM KWIC INDEX DISK VERSION 1
20 REM REF APPLICATIONS AND ALGORITHMS IN COMPUTER SCIENCE
30 REM BY C.W.GEAR PUBLISHED BY SRA 1978
40 REM TRS80 DOISK BASIC BY L.E.SPARKS
50 REM VARIABLES A$=AUTHOR,T$=TITLE,R$=REFERENCE
60 REM W$=KEY WORD,WI$=IGNORED WORD,NA=NUMBER OF ARTICLES
70 REM IN INDEX NUMBER,M=NUMBER OF TITLES ,N=NUMBER OF WORDS IGNORED
80 REM CLEAR STRING SPACE
90 CLS
100 CLEAR 10000
110 DEFINT I,J,K,L,M,N:DEFSTR A,R,W,T
120 INPUT "HOW MANY TITLES ARE TO BE INDEXED";M
130 DIM A(M+1),T(M+1),R(M+1),W(350),WI(110),IN(M+1),IT(350)
140 GOSUB 2020   REM INITIALIZE CHAINED LIST
150 IS=1
160 INPUT"ARE YOU ADDING TO A LIST STORED ON DISK",Y$
170 IF Y$<>"Y" THEN 270
173 INPUT"FILE NAME OF LIST OF ARTICLES",NR
175 OPEN"I",1,NR
180 REM NOW READ DATA FROM DISK
190 INPUT#1,NR
210 FOR I=1 TO NR
220     INPUT#1, A,T,R
230     PRINT A,T,R
240     GOSUB 2090  REM INSERT DATA FROM DISK INTO CHAINED LIST
250 NEXT I
260 CLOSE :REM CLOSE FILE
270 L=1
280 FOR I=1STON+1
290     CLS
300     INPUT"AUTHOR (END TO STOP)",A
305    IF A="END" THEN430
310     INPUT"TITLE ",T
320     INPUT"REFERENCE ";R
330     CLS:PRINT "AUTHOR ",A
340     PRINT "TITLE ";T
350     PRINT "REFERENCE ",R
360     INPUT"IS THIS CORRECT",Y$
370     IF Y$="Y" GOTO410
380     PRINT "REENTER INCORRECT DATA"
390     GOTO300
400     REM INSERT DATA INTO CHAINED LIST
410     GOSUB 2090
420 NEXT I
430 NA=NA+I-1
440 GOSUB 5000 :REM PRINT OUT  NUMERICAL LIST OF TITLES
450 GOSUB 2250 :REM PRINT OUT ALPHABETICAL BY AUTHORS LIST
460 REM ENTER WORDS TO BE IGNORED
465 INPUT"ARE YOU ADDING TO A LIST STORED ON DISK",Y$
470 IF Y$="N" THEN 550
475 INPUT" FILE NAME FOR IGNORED WORDS";NI
480 OPEN "I",1,NI
490 INPUT#1,WI
500 FOR I=1TON1
510     INPUT#1,WI(I)
520 NEXTI
530 CLOSE
535 N=N1
540 GOTO630
550 INPUT"HOW MANY WORDS TO BE IGNORED";N
560 II=1
570 FOR I= II TO N
580     INPUT"WORD TO BE IGNORED (ZZZ TO STOP)",WI(I)
590     IF WI(I)="ZZZ" THEN 610
600 NEXT I
610 N=I-1
620 CLS
630 PRINT"THE FOLLOWING WORDS WILL BE IGNORED, INDEX #"
640 GOSUB 1700
650 WI=N
660 INPUT"DO YOU WANT TO CHANGE ANY OF THESE WORDS",Y$
670 IF Y$="N" THEN GOTO 730
680 INPUT"INDEX NUMBER OF WORD TO BE CHANGED",IC
690 INPUT"ENTER NEW WORD ";WI(IC)
700 INPUT" ARE THERE ANY MORE WORDS TO CHANGE ";Y$
710 IF Y$="N" THEN 630
720 GOTO 600
```

```
730 INPUT"DO YOU WISH TO ADD TO THIS LIST";Y$
740 IF Y$="N" THEN 790
750 INPUT"HOW MANY WORDS DO YOU WANT TO ADD",NN
760 II=N
770 N=NN+II
780 GOTO570
790 REM NOW CONSTRUCT INDEX
800     M2=IN(1)
810     TX=I(M2)
820     TY=" "
830     GOSUB 1960
840     IF MI=0 THEN 910
850     W=LEFT$(TX,MI+LEN(TY)-2)
860     GOTO920
870     TX=RIGHT$(TX,LEN(TX)-LEN(W)-1)
880     GOSUB1960
890     IF MI=0 THEN  910
900     GOTO 850
910     W=TX
920 REM CHECK TO SEE IF WORD IS ON IGNORED LIST
930     GOSUB 2380
940     IF W="" THEN 960
950     IF MI<>0 THEN 870
960     M2=IN(M2)
970     IF M2=0 THEN 990
980     GOTO 810
990 CLS:REM OUTPUT INDEX
1000 TC=""
1010 LPRINT TAB(47)"  INDEX
1020 GOSUB 1580
1030 REM NOW PRINT INDEX
1040 FOR I=1TO L
1050     TX=I(IT(I))
1060     TY=" "
1070     GOSUB 1960
1080     IF MI=0 THEN  1140
1090     TW=LEFT$(TX,MI+LEN(TY)-2)
1100     TX=RIGHT$(TX,LEN(TX)-LEN(TW)-1)
1110     IF W(I)=TTW THEN 1190
1120     TC=TC+" "+TW
1130     GOTO 1070
1140     TW=TX
1150     IF MI=0 THEN 1170
1160     TC=-TC+" "+TW
1170     IF TW<>W(I) THEN 1270
1180     TX=""
1190     Z=40-LEN(TC)
1200     IF Z<0 THEN Z=0
1210     TC=STRING$(Z," ")+TC
1220     IF MI=0 THEN TX=""
1230     TX=TW+" "+TX
1240     X=30-LEN(TX):IF X<0 THEN X=0
1250     TX=TX+STRING$(X," ")
1260     LPRINT TC;TX,"          ",IT(I)
1270 NEXTI
1280 INPUT"DO YOU WANT TO SAVE EVERYTHING ON DISK";Y$
1290 IF Y$="N" THEN END
1300 CLS:INPUT"FILE NAME FOR ARTICLE LIST ";WA
1310 OPEN "O",1,WA
1320 INPUT"FILE NAME FOR KEY WORD LIST ";WK
1330 OPEN "O",2,WK
1340 INPUT"FILE NAME FOR IGNORED WORD LIST ";WI
1350 OPEN "O",3,WI
1360 PRINT #1,NA
1370 FOR I=2TONA+1
1380     PRINT#1,A(I);",";T(I);",";R(I)
1390     PRINT A(I),T(I),R(I)
1400 NEXT  I
1410 PRINT#2,L
1420 FOR I=2TOL
1430     PRINT #2,W(I);",";IT(I)
1440     PRINT W(I),IT(I)
1450 NEXTI
1460 PRINT#3,NI
1470 FORI=1TONI
1480     PRINT#3,WI(I)
1490     PRINT WI(I),
1500 NEXT I
1510 PRINT
1520 CLOSE
1530 REM NOW HAVE FINISHED
1540 END
1580 REM SHELL SORT SUBROUTINE FOR KEY WORDS
1590 IM=L-1
1600 IM=INT(IM/2)
1610 IF IM=0 THEN 1760
1620 J=1
1630 K=L-IM
1640 I=J
1650 IL=I+IM
1660 IF W(I)<W(IL) THEN 1730
1670 TE=W(I):IE=IT(I):REM TEMPORARY STORE FOR KEY WORD AND INDEX
1680 W(I)=W(IL).IT(I)=IT(IL).REM SWAP PLACES
```


✓ Reader Service—see page 147

```
1690 W(IL)=TE IT(IL)=IE
1700 I=I-IM
1710 IF I<1 THEN 1730
1720 GOTO 1650
1730 J=J+1
1740 IF J<=K THEN 1640
1750 GOTO 1600
1760 RETURN
1780 REM BUBBLE SORT FOR IGNORED WORDS
1790 N1=N
1800 S=0
1810 N1=N1-1
1820 FOR J=1TON1
1830    IF W1(J)<=W1(J+1) THEN 1860
1840    W1=W1(J) : REM TEMPORY STORE FOR W1(J)
1850    W1(J)=W1(J+1) REM SWAP PLACES
1860    W1(J+1)=W1
1870    S=1
1880 NEXT J
1890 IF S=1 THEN 1800
1900 REM NOW PRINT LIST ON CRT
1910 FOR IX=1TON
1920    PRINT W1(IX),IX,
1930 NEXT IX
1940 PRINT
1950 RETURN
1960 REM INSTRING ROUTINE REF TRS80 LEVEL II MANUAL
1970 FOR  M1=1TO LEN(TX)-LEN(TY)+1
1980    IF TY=MID$(TX,M1,LEN(TY)) RETURN
1990 NEXT M1
2000 M1=0 RETURN
2010 REM SUBROUTINE TO INITIALIZE CHAINED LIST POINTER
2020 M2=2 REM M2 IS POINTER
2030 IN(1)=0
2040 FORI=2TOM+1
2050    IN(I)=I+1
2060 NEXT I
2070 IN(M+1)=0
2080 RETURN
2090 REM SUBROUTINE TO INSERT DATA INTO CHAINED LIST
2100 REM M2 IS  POINTER TO FREE SPACE
2110 IF M2=0 THEN 2240
2120 M1=1
2130    IF IN(M1)=0 THEN 2170
2140    IF A(IN(M1))>A THEN 2170
2150    M1=IN(M1)
2160 GOTO2130
2170 REM NOW INSERT DATA INTO LIST
2180 J=M2
2190 M2=IN(J)
2200 IN(J)=IN(M1)
2210 A(J)=A T(J)=T:R(J)=R
2220 IN(M1)=J
2230 RETURN
2240 PRINT" NO FREE SPACE ".RETURN
2250 REM SUBROUTINE TO PRINT ORDERED LIST
2260 LPRINT"ALPHABEETICAL BY AUTHORS LIST OF  ARTICLES"
2270 LPRINT"PRINTED AS FOLLOWS REF#, AUTHOR; TITLE, REFERENCE"
2280 LPRINT" "
2290 M2=IN(1):REM SET POINTER AT START
2300 REM DOO WHILE M2<>0
2310    LPRINT M2;"  ";
2320    LPRINT A(M2),"  ",
2330    LPRINT T(M2);"   ";
2340    LPRINT R(M2)
2350    M2=IN(M2):REM SET POINTER TO NEXT ITEM
2360    IF M2=0 RETURN
2370 GOTO2300
2380 REM BINARY SEARCH TO SEE IF WORD IS ON IGNORED LIST
2390 K=0
2400 IL=1
2410 IH=N
2420    IF IL>IH THEN 2500
2430    IM=INT((IL+IH)/2)
2440    IF W1(IM)=W THEN RETURN
2450    IF W1(IM)>W THEN 2480
2460    IL=IM+1
2470 GOTO 2420
2480    IH=IM-1
2490    GOTO2420
2500 W(L)=W:REM SAVE WORD BECAUSE IT IS NOT ON IGNORED LIST
2510 IT(L)=M2:REM KEEP TRACK OF REF#
2520 L=L+1
2530 RETURN
5000 REM SUBROUTINE TO PRINT NUMERICAL LISTING
5010 LPRINT" LIST OF ARTICLES"
5020 LPRINT" "
5030 FORI=2TONN+1
5040    LPRINT I,T(I),T(I)
5050    LPRINT TAB(15)R(I)
5060 NEXTI
5070 RETURN
```

I made the TRS-80* into a serious computer. Now I've made the Model II into a spectacular one.

I'm Irwin Taranto, and I've helped almost a thousand businesses get their first computers up and running.

I've done it primarily with the TRS-80, because it's a really elegant piece of hardware. Given the right programs, it can do substantially the same work as the traditional minicomputers that cost four times as much.

I proved it with four on-line, interactive programs adapted from the genuine Osborne & Associates systems, originally designed for the $30,000 Wang computer. Then I added two of my own and made them all work on a $4000 TRS-80.

Now I've done the same thing for the new TRS-80 Model II. It's an $8000 computer that works twice as fast and has four times the memory—up to two million characters.

My new systems are fully documented, and because I'm working with a much more powerful computer, they're a night-and-day advance over the Model I programs. They'll turn your Model II into a complete business computer, set up and ready to go.

THE TRS-80 MODEL II PROGRAMS

General Ledger/Cash Journal: handles up to 7000 transactions on 500 different user-defined accounts. It keeps track of them by month, quarter and year, makes comparisons to the prior year, and does departmentalization.

Accounts Payable/Purchase Order: generates the purchase order and posts the item to payables when the goods are received. Invoice-linked, it calculates and prints checks and aged ledger reports and links fully to the general ledger.

Accounts Receivable/Invoicing: keeps track of billed and unbilled invoices, open and closed items, aging and service charge calculation. It prints statements, links to the general ledger, and can work within either an invoice-linked or balance-forward accounting system.

Payroll/Job Costing: computes regular, overtime and piecework pay, keeps employee files, figures taxes and deductions, prints checks, journal, 941-A and W-2 forms, and breaks out individual job costs.

*A trademark of the Tandy Corporation.

When I say set up and ready to go, I mean just that. If you're not quite sure on that point, call the number below and we'll give you the names of some of the people who've already bought all over the world. Call them up and hear what they have to say.

These Model II programs are completely custom-tailored, which explains their $249.95 price. Before we'll send you a disk, you have to fill out a detailed questionnaire that tells us your precise business requirements. Then we send you the disk, all the instructions you need, and my phone number. If you call, we answer all your questions. If your questions are tough enough, I'll talk to you personally.

Because that way I'll make sure that Model II of yours turns into a spectacular computer, just like I promised.

☐ Please send me the custom questionnaires for the following $249.95 Model II programs:

- ☐ General Ledger/Cash Journal
- ☐ Accounts Payable/Purchase Order
- ☐ Accounts Receivable/Invoicing
- ☐ Payroll/Job Costing

☐ Please send me information on the TRS-80 Model I programs at $99.95 each

☐ Please send me information on other Taranto business programs

Your name _____

Company name _____

Address _____

City/State/Zip _____

✓45

Taranto
& ASSOCIATES, INC.

P.O. Box 6073, 4136 Redwood Hwy., San Rafael CA 94903 • (415) 472-2670

You can have the best of two worlds with this software driven modification.

lowercase and UPPERCASE

Donald L. Stoner
Richard Berker
The Peripheral People
PO Box 524
Mercer Island WA 98040

Radio Shack didn't overlook much when it designed the TRS-80 system. It is unquestionably the most popular computer of all time, with sales well into six figures.

At the time the TRS-80 was being developed, graphics was the big buzzword in the hobby computer industry. Graphics helped in the transition from video games to hobby computers for the American public. The marketing people at Tandy were probably so insistent on having better graphics that they overlooked one of the most important markets for the TRS-80—word processing.

If you are frugal and have a bit of electronic knowledge, you can build up a word processor for around $1000. The basic TRS-80 costs $600, and the careful shopper can find plenty of Selectrics for less than $400.

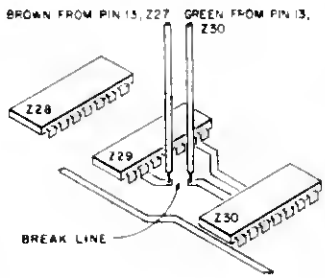


Fig. 1a.

It's impossible to have an effective word processor without an uppercase/lowercase capability, however. An early brochure on the TRS-80 mentioned that you could have uppercase and lowercase in your TRS-80, but you would have to give up the graphics capability. Despite sending repeated inquiries (some of them heated) to Tandy Corp., I failed to elicit exactly how this could be done.

As it turns out, you can easily have keyboard selection of uppercase or lowercase without giving up graphics. The information that follows tells you how. I call it a "convertible conversion." It is simple to install but, more important, can be removed in a matter of minutes in case you need warranty repairs to your keyboard. No holes are drilled in the case or circuit board.

Materials

Besides the usuel tools end soldering iron, you will need the components shown in Teble 1. All these materials are available from your Radio Shack store. Their part numbers are shown in parentheses. You will also need some electronic knowledge to complete the conversion. If you are a little weak in this area, consult a friend you consider knowledgeeble in electronics. He can be helpful if you get in trouble!

A switch is required in case you want to return the circuit to its original configuration. This is necessary if you own (or plan to purchase) any machine-language programs such as Micro-Chess 1.5. Without the switch, the alpha characters in Micro-Chess (and similar programs) appear as weird control characters. However, the switch is normally left in the conversion position and does not affect BASIC programs.

short lengths of Kynar wire (278-503)
one type 2102 IC (276-2501)
one type 7486 IC (276-1827)
one 4.7k, ¼ Watt resistor (271-030)
one DPDT toggle switch (275-614)
five 6 inch lengths of hookup wire (see Fig. 2)

Table 1. Parts list.

Conversion Procedure

Ready for the big step? Start by forgetting you paid more than $600 for the TRS-80 and plunge ahead.

1. Disconnect your keyboard and lay it face down on a bath towel to prevent scratching.

2. Remove the six screws. Note there are three different types. Be sure to get them back in the correct holes when reassembling.

3. Carefully turn the case over and remove the top cover. Lift out the keyboard assembly from the posts and remove the plastic spacers. Remove the second circuit board and set the bottom case section aside. Do not flex the copper cable (which connects the two boards) excessively. The two boards do not disconnect from each other.

4. Set the two boards down, component side up (with the keyboard to the rear), on your work towel. The values marked on the main board should read correctly (not upside down).

5. Observe the lower left area of the main circuit board (not the one with the keys). Locate IC chips Z60 and Z61. You will be piggybacking a new IC on top of each one of these chips. It you do not know how, learn to read the pin numbers of these integrated circuits. Pin 1 is the reference pin located at the upper left-hand corner of the chip, nearest the Z number printed on the circuit board. There is also a dot indentation in the plastic body of the chip nearest pin 1.

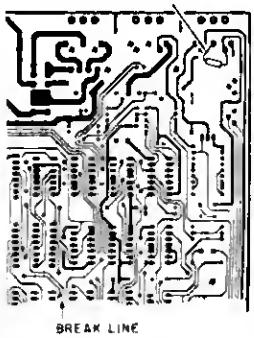


Fig. 1b.

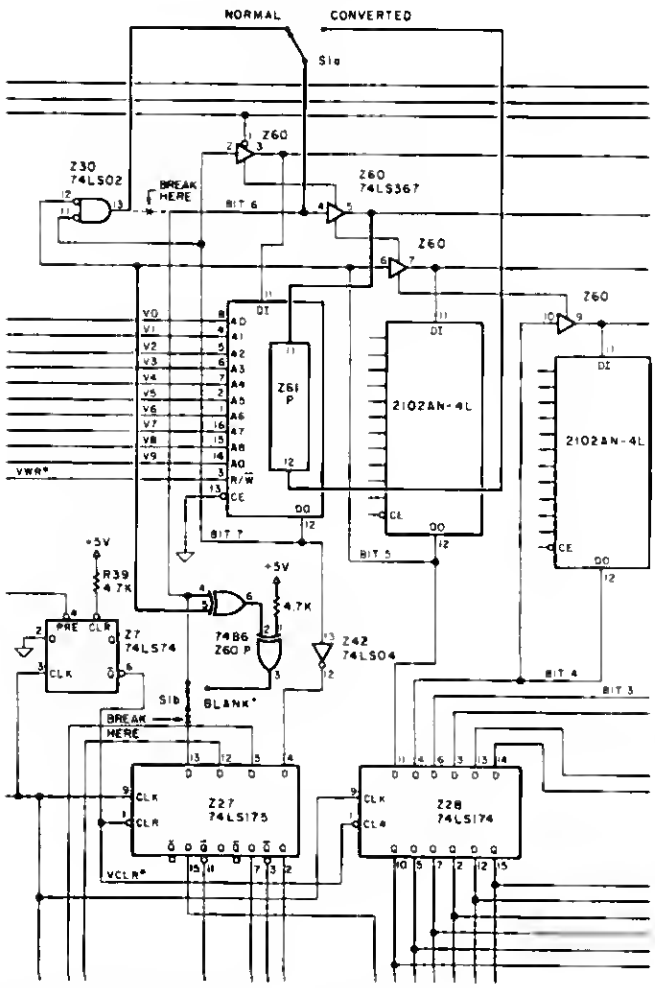


Fig. 3. Circuit diagram of the converted uppercase/lowercase TRS-80. *(Courtesy of Radio Shack)*

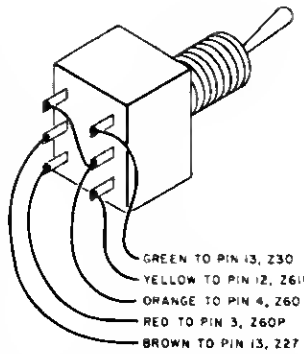


Fig. 2. Pre-wire the switch as shown here. Don't forget to add the jumper.

The pins are numbered successively from pin 1 down one side and up the other. Thus the highest-numbered pin (usually 14 or 16) is opposite pin 1. Note that Z60 (marked 74LS367) and Z61 (marked 2102) are both 16-pin chips. Also locate Z30. You are actually going to do a coronary bypass by cutting a circuit board trace near this chip and another circuit trace on the bottom of the board.

Still game? It is not too late to put everything back together, get some orange model airplane cement (butyl acetate), touch up the warranty paint seal, and Radio Shack will be none the wiser! Proceed, you say? Stout fellow!

6. We are going to stack the new 2102 chip on top of Z61, which also happens to be a 2102. First, however, bend up pins 11 and 12 of the new 2102 (let's call it Z61P, for piggyback) at right angles so they cannot touch pins 11 or 12 of Z61.

7. Next, solder (pin for pin and don't get it reversed end for end) Z61P on top of Z61. Use extreme caution to get all pins (except 11 and 12, of course) securely connected. Equally important, do not get any solder bridges between pins or from one of the pins to the circuit board.

8. Connect a short, direct wire from pin 5 of Z60 to pin 11 of Z61P. Pin 12 of Z61P will be connected to the toggle switch leter.

OK, that was the easy part. Next we have to piggyback the 7486 chip on top of Z60. We'll call this added chip Z60P. Unfortunately, Z60P is a 14-pin chip, while Z60 has 16 pins. Thus, we cannot make a pin-for-pin connection as we did with the 2102.

9. Bend pins 8, 9, 10, 11, 12 and 13 of Z60P at right angles so that they cannot touch the pins of Z60. Only pin 14 on this side of the chip will be used.

10. Bend pins 1, 14 and 7 slightly so they will contact 1, 16 and 8 when Z60P is joined to Z60.

11. Bend pins 4 and 5 of Z60P away from each other so they will contact pins 4 and 6 of Z60 when the two chips are joined.

12. Bend pins 2, 3 and 6 of Z60P at right angles so they cannot touch the pins of Z60.

13. Place Z60P over Z60 to ensure you can make the following solder connections. Look good? OK, solder the connections. The first number is Z60P; the second is Z60.

Pin 1 to Pin 1
Pin 14 to Pin 16
Pin 7 to Pin 8
Pin 5 to Pin 6
Pin 4 to Pin 4

14. Connect the 4.7k, 1/4 Watt resistor between pins 1 and 14 of Z60P.

15. Connect a short, direct length of Kynar wire from pin 2 to pin 6 of Z60P. For the moment, leave pin 3 of Z60P disconnected. It will be connected

Connect the green wire to pin 13 of Z30. Connect the orange wire to pin 4 of Z60. Connect the yellow wire to pin 12 of Z61P. Connect the red wire to pin 3 of Z60P. Connect the brown wire to pin 13 of Z27.

Table 2. Connecting the switch.

```
BF60:  F5 3A 18 40 FE 01 20 06 79 C5 CD 3B 00 C1 3A 19
BF70:  40 FE 01 28 04 F1 C3 58 04 F1 DD 6E 00 DD 66 04
BF80:  DA 9A 04 DD 7E 05 B7 28 01 77 79 FE 80 D2 A6 04
BF90:  FE 20 DA 06 05 FE 40 DA 7D 04 FE 60 30 05 F6 20
BFA0:  C3 7D 04 E6 9F C3 7D 04 3A 19 40 FE 01 20 14 79
BFB0:  FE 41 38 0E FE 7A 30 0A FE 58 38 04 FE 20 38 02
BFC0:  EE 20 4F 3A 1A 40 FE 01 C2 8D 05 79 FE 0D 28 05
BFD0:  FE 0A C2 8D 05 11 00 20 1B 7A B3 20 FB 3E 0D 32
BFE0:  E8 37 11 00 20 1B 7A B3 20 FB 3E 0A 32 E8 37 11
BFF0:  00 30 1B 7A B3 20 FE 0E 0D C9 00 00 00 00 00 00
```

Listing 1.

to the toggle switch later.

16. Refer to the sketch in Fig. 1 (a). Locate the area shown between Z29 and Z30 and cut the trace with an X-acto knife at the point shown. This point can be bridged with a short piece of bare Kyner wire if it is necessary to remove the conversion.

17. Similarly, on the reverse side of the board, locate the circuit trace that goes from pin 13 of Z30 to pin 4 of Z60 (see Fig. 1 (b)). Make a small cut in this trace that can be bridged later, if the conversion is removed.

18. The last conversion step is to connect the toggle switch. Use the 6 inch lengths of hookup wire to prepare the switch as shown in Fig. 2. Connect the switch as directed in Table 2. Don't forget to include the jumper on the switch, as shown in Fig. 2. Place the switch in the conversion position (with the handle toward the green wire end).

This completes the case conversion of the TRS-80. Carefully review your work to make absolutely certain there are no shorted wires or solder bridges. Check things with an ohmmeter if you have any doubt about any connection. Once you are certain all is well, reassemble everything by reversing the disassembly steps done earlier. Route the cable (with the toggle switch on the end) out the hole where the interface plug connects.

Power Up

After reconnecting everything, power up the system normally. The screen display should be the same as before with one small exception. Your cursor will no longer be a dash but, rather, will look like ▯. This is one of the control characters mentioned earlier. It takes some getting used to, but pretend it's a happy face!

Despite all that work, we still have no lowercase letters. For this, you are going to have to enter some software. If you did not purchase a software tape from The Peripheral People, use your T-BUG or DOS to enter the machine-language program in Listing 1. It is relocatable, depending on how much memory you have.

The BF is the location. It should be 7F, BF or FF for 16K, 32K or 48K, respectively. Don't forget to protect your memory at 32605, 48991 or 65375, or your programs will go crashing into this routine with disastrous results. Incidentally, don't forget to also save the routine on tape or disk. Having to enter the machine code each time you want to use the uppercase-lowercase conversion can become messy.

```
POKE16414,96:POKE16415,XXX:POKE16422,167:POKE16423,XXX.
```

Example 1.

```
INPUT"WOULD YOU LIKE UPPER AND LOWER CASE";A$
IF LEFT$(A$,1) = "Y" THEN POKE &H4019,1 ELSE POKE &H4019,0
```

Example 2.

Testing

To see if all your hard work and electronic expertise paid off, type and enter the patch (in Example 1) to the start of the routine. The XXXs are the starting address and are the decimal equivalent of 7F, BF or FF for 16, 32 and 48K, respectively. This patch will be required at the start of any program that requires uppercase and lowercase. Next, type and enter &H4019,1.

The screen should show READY in lowercase letters. Besides this, the first thing you will probably notice is that the letter a is sitting above the baseline of the words. Early TRS-80s had a Motorola character generator ROM with an error in the font for this letter. In later units, this ROM error was corrected.

You will also find that the tails on letters such as p, q, y and so on don't extend below the baseline. This is because the character generator is only a 5 x 7 matrix. There simply are not enough dots available to print the tails below the baseline. The letters could be shifted electronically, but this is hardly worth the complication, trouble and expense (translation: I don't know how). Once you get used to the shifted letters, you won't notice them anyway.

You can return to uppercase only by typing POKE &h4019,0. The two POKE statements can be built into your program (see Example 2).

It is interesting that an unmodified TRS-80 does have uppercase and lowercase printing capability. Naturally, this is only apparent with a printer having a lowercase capability. However, LPRINT produces uppercase printing all the time unless you use the shift key when writing the program. Unfortunately, pressing the shift will cause lowercase letters to be printed, which is just the opposite of what you want.

The Program

The program which accompanies this article has provision for reversing the case out the printer port. The case reversal to the printer is made automatically when you use the above POKE statement. Naturally, when you POKE back to zero, the normal unshifted uppercase printing occurs.

The accompanying program also has provision for inserting an automatic line feed with each carriage return. If you have a Teletype or similar machine that requires this, simply POKE &H401A,1 to turn it on and &H401A,0 to turn it off. The program will even add a carriage return after a line feed as required by some printers.

Finally, the program has an echo routine. To the best of my knowledge, this is the first time this feature has been offered to TRS-80 owners. This is extremely handy for a couple of reasons. Let's say you have a number of PRINT statements that must be changed to LPRINT before the printer will work. You can type POKE &H4018,1 just before these statements. Any subsequent print statements will echo on the screen and printout without adding L ahead of each.

The echo is also handy to turn your printer into a typewriter. By entering the echo POKE statement, anything you type on the keyboard will echo on the printer. When you want to turn off this feature, simply type POKE &H4018,0.

These features are all incorporated in the Electric Secretary word-processing program, which was used to type this manuscript. It is available from The Peripheral People for $50 postage paid on a customer-supplied DOS-formatted disk for the TRS-80. The Electric Secretary can also be supplied on cassette for the same price, for customer transfer to a disk. Note, however, it is a disk-based system only. Because of the self-contained hyphenating dictionary (and other features) a cassette-based system is not fast enough. A complete conversion kit of parts, including a machine-language program tape, is also available from The Peripheral People (Box 524, Mercer Island WA 98040) for $20 postage paid. ■

Seems that a label can cost more than $100, read why.

Quick Printer

Henry G. Riekers
208 Phelps Avenue
Glen Burnie MD 21061

Centronics Data Computer Corporation of Hudson NH manufactures the P1 Microprinter, which is also known as the Quick Printer when sold under the Radio Shack label.

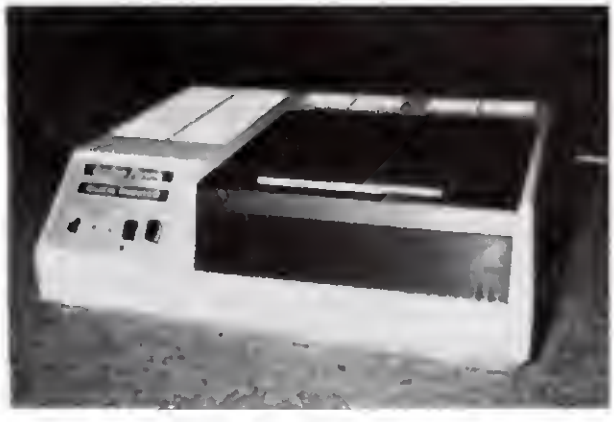
The P1 is a seven-bit ASCII TTL printer with strobe and acknowledge pulse that employs nonimpact discharge technology that requires only four moving parts to produce variable-pitch 5 × 8 dot matrix characters at a rate of 150 lines per minute, with a vertical density of 5 lines per inch. The paper, which is 4.75 inches wide, carries a conductive aluminized coating that is vaporized by a low voltage discharge from the printhead. Printed characters are highly legible, and excellent copies can be made.

A number of software commands initiated by the TRS-80 provide the user with considerable flexibility, such as printing at either 5, 10 or 20 horizontal characters per inch. Underlining may be started and stopped by separate commands. An audio alarm, also under software control, provides a loud two-second tone. The full 96-character ASCII set, including both upper and lowercese letters, can be printed.

Connection of the P1 to the TRS-80 is accomplished through a Radio Shack 26-1401 cable end the Radio Shack 26-1140 Expansion Interface. (Note: Radio Shack has recently announced a lower-priced alternative for the connection of parallel printers to the TRS-80 without the need for the Expension Interface.) No electrical changes are required to the Centronics P1 printer to make it compatible with the TRS-80.

The Centronics manual provides the octal software codes for printer control. Since the TRS-80 initiates these commands in decimal format, they must be converted by the operator. Thus, LPRINTCHR$(29) prints 20 characters per inch; LPRINTCHR$(30) prints 10 charecters per inch; and LPRINTCHR$(31) prints 5 characters per inch, which is typed into the computer, for exemple, as LPRINTCHR$(31) "Radio Shack." Underlining is started and stopped using LPRINTCHR$(15) and (16), respectively. The audio alarm is sounded by LPRINTCHR$(07).

The commands for print size may be given prior to listing a program or may be included in the body of the program to ob-



The Quick Printer.

The Centronics P1.

tain various printing effects. In addition, lowercase letters may be printed directly on the P1 by depressing the shift key when a program is typed on the keyboard. The printing will still appear as all uppercase on the CRT, but will be both upper and lowercase on the P1 for whatever printing density is selected.

Since each character can be printed using its ASCII code in the CHR$ format, it is possible to print characters not otherwise possible, such as quotation marks. In addition, printing may be stopped and started, paper advanced and printhead positioned.

The P1 is 13 inches wide, 10.5 inches deep, 4.25 inches high, weighs 10 pounds and consumes 40 Watts while printing. The case is made of rugged plastic decorated in ivory and black. Controls on the front of the printer consist of a power switch to turn the unit on and off, a select switch that allows data to enter the printer and a paper-feed button that permits the operator to advance the paper. Even though power is "on," the motor turns off when no data is present.

Indicators consist of lamps displaying power "on" and "paper empty," which also sounds an audio alarm. Paper is available from Radio Shack or directly from Centronics in either shiny or matte aluminum finish. The printer is not supplied with a mating connector.

The Centronics P1 is available through many supply houses for $395 each. The Quick Printer is sold by Radio Shack for $499 and is available off-the-shelf in many stores. ■

MICROPRINTER
MICROPRINTER
MICROPRINTER

Microprinter
Microprinter
Microprinter

Sample output from P1.

You too can earn money with your computer expertise.

Part-Time Consultant

Michel Morin
15 Jasper Rd.
Candiac Quebec
Canada J5R 4E7

I am an audio-visual technician and a computer hobbyist who originally bought a TRS-80 to play Star Trek and develop a system to play music with an organ. I have since become a part-time consultant on the TRS-80 Business System. This article is a description of a recent project in which I am still involved.

Background

In October 78, a local Radio Shack store manager asked me if I was willing to help two potential customers who had a problem that might be solved by a TRS-80 Business System. These two customers operated school buses and other inter-urban buses and wanted to produce a set of specialized and complex reports. Fifteen of these reports were to be completed yearly and sent to the provincial government. They were to be based on data that the bus operators hed never compiled in the past.

After studying their problem and the specs of the Radio Shack System, I accepted the challenge. Both of them placed orders for a 32K Business System with two disk drives and a printer.

At that time, Radio Shack Canada was taking orders for disk based systems, but nobody had ever seen one. My first move was to get, by nearly fraudulent means, a copy of the DOS manual that was to come with the system. Until then, I had had no experience with Disk BASIC.

I deciphered the bus company's reports and determined that I would have to put the whole accounting of the business on the system. Since I didn't know the first thing about the subject, I found some help in the form of a younger brother who's an accountant.

Our first decision was to write a General Ledger program to which we could add others as the need arose. The ledger would need 235 accounts and most of these had to appear six times. We created the flow charts from scratch and wrote a few program lines, while waiting for Radio Sheck peripherals.

Finally, in January 79, the disk drives arrived at the store and I think they were the first ones in the Montreal region.

Problems

The first problem we had, because of our lack of experience and problematic instructions from Radio Shack was setting up the system. We thought we had the system working and decided to make a couple of backups of the DOS just in case. Alas, Radio Shack didn't have diskettes to sell in Canada, so we had to buy some Verbatim at an outrageous price in the only computer shop in Montreal.

The GL accounts would be kept in a random file and the daily transactions in a sequential file which we'd scrap each month, after producing a detailed journal of accounts.

The DOS (version 2.1) started acting up and on at least three occasions zapped all of our files, while committing suicide itself. After much searching, we found that one of the disk drives had tracking problems and luckily we found a replacement for it.

We had problems each time the drives were trying to write or read a sequential file and resolved to avoid using them and to post transactions as they were entered in the ledger.

Another problem was time. One of our programs that generated a beautiful report took about 12 hours to run because of certain characteristics of Disk BASIC using sequential files. We scrapped that report and produced a summary of transactions for each session at the computer that took a lot less time.

Our handicap in these early attempts was Radio Shack's version 2.1 DOS.

Right now, our customers use the programs daily and they work just fine. Since each transaction is posted as entered, they can always generate up-to-date financial reports and they love that!

We are currently working on a set of programs that will keep track of data pertaining to the operation of each vehicle, which, when combined with the General Ledger will yield all the statistical information desired by the government.

Conclusion

This initial experience has led me to other customers with special applications problems for which I write programs. If a customer has standard applications, I encourage him to buy the Osborne programs sold by Taranto or Computronics and I translate them into French for his system.

Why the Osborne programs? Because Radio Shack is unable to fill its customer orders. My first customers are still waiting for Radio Shack's Canadian payroll program, first promised 1½ years ago. ■

If you've installed some additional memory and want to put it to the test—read on.

Test Your Memory

Milan D. Chepko
119 Belleville Court
Thief River Falls MN 56701

After adding a disk drive to my 16K TRS-80 system, I began to feel a "memory crunch" because of the DOS and DISK BASIC tying up some of the RAM. Though Radio Shack will cheerfully install another set of 16K RAM chips for $200, this seems a little steep, especially now that several companies sell kits of the same chips for around $70.

Installing the chips in the Expansion Module was relatively easy (why do the first set of chips go in sockets Z9-Z16 while the second set go in sockets Z1-Z8?), but later I began to have some nagging doubts about how good my new memory really was. I had some bad experiences with my old 4K RAM boards using 2102s a few years ago, so I decided to check out the new memory completely before entrusting my programs to it.

First, I reviewed the memory structure of the TRS-80 system. As shown in Fig. 1, the total addressable memory space is broken into two blocks of 32K, the first half residing in the CPU/keyboard and the second half in the Expansion Module.

In the CPU, the first 16K is used for the ROM chips holding Level II BASIC, the video RAM chips and the memory space allotted to the keyboard itself.

The second 16K is made up of eight RAM chips and is generally available for programming (although several hundred bytes at the beginning are used by the processor for "housekeeping" tasks, so only 15K or so is really usable). PEEK and POKE can be used to address any of these locations, which are numbered from 0 to +32767 in digital format.

In the Expansion Module, there is space for a total of 32K of RAM in two banks of eight memory chip sockets. Each bank of eight sockets provides 16K of RAM when the chips are inserted. The addresses (in digital format) for this second half of memory run from −32768 to −1, which causes some confusion at first. The best approach is to figure that the location just after +32767 is −32768, the next is −32767 and so on.

Testing the Memory

Testing the memory turned out to be easier than figuring out the addresses! Essentially, all memory tests consist of storing a specific bit pattern at a location, reading it back and comparing the result with what should be there. If the results are different, an error message is generated.

No memory test could ever cover all the possible bit patterns in a reasonable running time, so some shortcuts must be used. One effective method is to store alternating bit patterns like 10101010 and 01010101 in sequential memory locations, then check to see if any have been forgotten or disturbed by their neighbors. If the pattern at any location is not as expected, there is a defect either in a memory chip or in the traces on the board.

While it is helpful to know which bit is incorrect, it is even more useful to know which memory chip is at fault. Each of the eight bits in a memory location resides in different memory chips, lying side by side on the board. By determining which bit is incorrect, we narrow the problem down to a specific memory chip.

Once a bad chip is identified, it is always a good idea to swap it for one of the known good chips on the board and run the memory test again. If the defect moves with the chip, pack it up and send it back to the supplier. If the defect stays at the original location, the chip is probably OK, but there may be something wrong with the board itself—possibly a solder bridge that you can find and remove with a little searching.

The program contains its own instructions along with REM statements to describe the function of each section. After determining the locations to be tested, the program fills them with the alternating bit pattern. It then goes back and checks first for one pattern and then the other.

The pattern is reversed for the second pass through memory, and whenever a defect is encountered the program branches to e subroutine that determines which chip is at fault.

There is provision for use of a line printer, which can be very

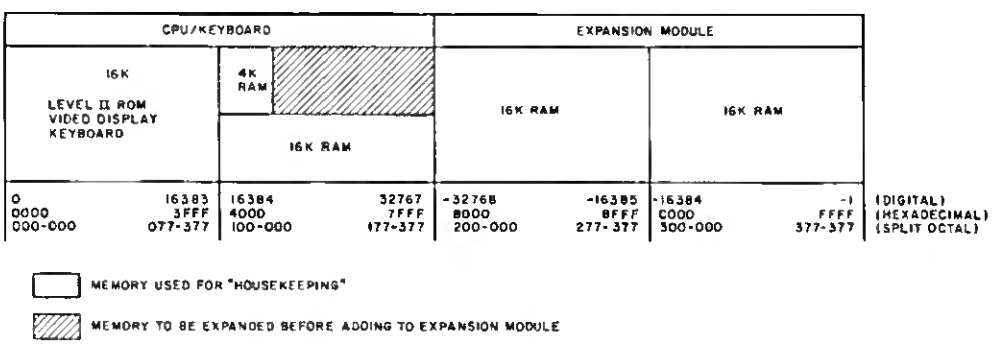
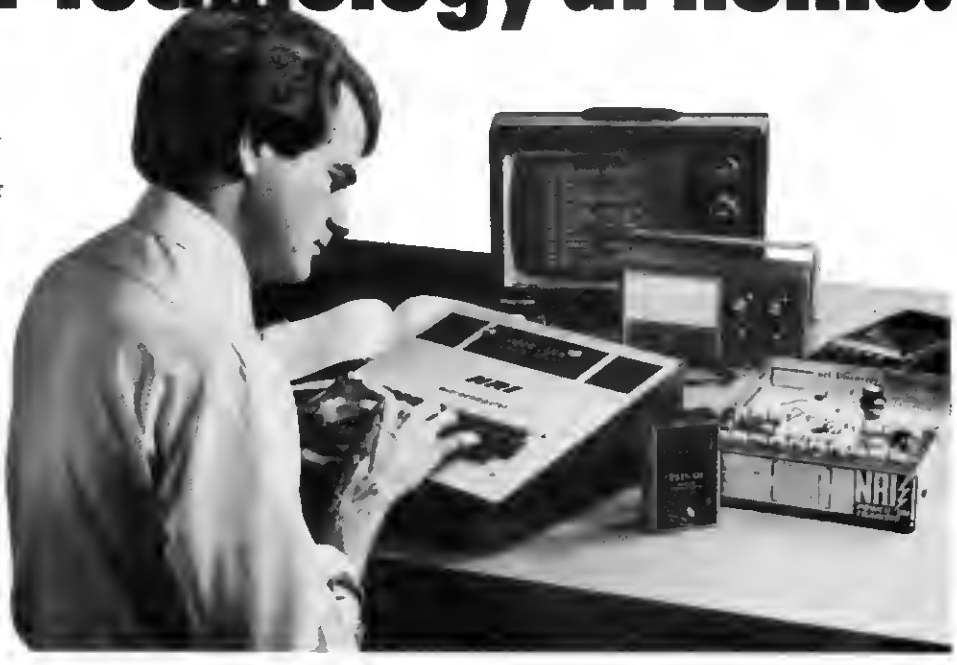


Fig. 1

Build your own microcomputer as you learn computer technology at home.



New from NRI! The First Interdisciplinary Home Study Course Ever Offered

As the microprocessor revolutionizes the computer world and microcomputers appear almost everywhere, NRI brings you a new, convenient, and effective way to keep up with this expanding technology. It's NRI's courses in Microcomputers and Microprocessors, created and designed exclusively for learning at home in your spare time.

Designed for the New Breed of Computer Technician

It's no longer enough to be just a programmer or technician. With microcomputers moving into the fabric of our lives as low-cost, easily available tools for business and home, both the programmer and technician must become total professionals. With practical knowledge of hardware, the programmer can design simpler, more effective programs. And with advanced programming skills, the technician can test and debug systems quickly and easily. The NRI course gives you simultaneous training in both skills...makes you one of this rare new breed.

Build Microcomputer, Test Instruments

NRI goes far beyond book learning to give you practical, "hands-on" experience. As you learn, you actually assemble NRI's designed-for-learning microcomputer. It performs like the finest of its kind, and features both assembly and basic language capabilities.

Every assembly step's a learning step. Using the NRI Discovery Lab® plus the NRI transistorized volt-ohm meter and CMOS digital frequency counter you also build, you perform meaningful experiments throughout your course...trace circuitry, interface components, introduce and correct problems, design your own programs, and more.

The Proven Way to Learn at Home

You don't have to worry with travel, classes, or time lost from work when you learn the NRI way. As they have for more than 60 years of teaching technical subjects, NRI brings the material to you. You study in your spare time, at your convenience, using "bite-size" lessons that program material into logical segments for easier assimilation. You perform experiments and build equipment using kits we supply. And your personal NRI instructor is always available for consultation should you have questions or problems. Over a million students have already shown the effectiveness of NRI training.

Choice of Courses

Several courses are available, depending upon your needs and background. NRI's Master Course in Microcomputers and Microprocessors starts with the fundamentals, explores basic electronics and digital theory, the total computer world, and the microcomputer. The Advanced Course, for students already versed in digital electronics, concentrates on software and the world of the microprocessor and microcomputer. In both courses, you build all instruments and your own computer.

Send for Free Catalog... No Salesman Will Call

Get the details on these exciting new courses in NRI's free, 100-page catalog. Shows all kits and equipment, lesson outlines, and full information, including facts on other electronics courses. Mail the coupon today and we'll rush your catalog. No salesman will ever call. Keep up with the latest technology as you learn on your own computer. If coupon has been removed, write to NRI Schools, Computer Department, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI McGraw-Hill

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

NO SALESMAN WILL CALL
Please check for one free catalog only.

All career courses approved under GI Bill.
☐ Check for details.

- ☐ Computer Electronics Including Microcomputers
- ☐ TV/Audio/Video Systems Servicing
- ☐ Complete Communications Electronics with CB • FCC Licenses • Aircraft, Mobile, Marine Electronics
- ☐ CB Specialists Course
- ☐ Amateur Radio • Basic and Advanced
- ☐ Digital Electronics • Electronic Technology • Basic Electronics
- ☐ Small Engine Repair
- ☐ Electrical Appliance Servicing
- ☐ Automotive Mechanics
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Refrigeration, & Heating Including Solar Technology

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council

179-030

✓ Reader Service—see page 147

useful if more than two or three defects are present. The program directs all output to the screen unless the printer is connected and turned on, in which case POKE statements trick the processor into sending the output to the line printer. The use of PRINT" " causes a line feed regardless of whether the TV or line printer is being used.

Conclusion

The only possible problem is in using the negative numbers to address the memory locations being tested, as explained above.

Also, if you are testing a full 16K block, be prepared to wait about 10 minutes. I've found it helpful to test for one location beyond what I know is there (−16384), so that there is always an error statement generated at the end of each checking routine. Above all, be sure that you protect the Expansion Module memory by entering '32767' for MEMORY SIZE, or the program will try to store variables in the same memory you are testing!

In case you have been thinking about testing the 16K of RAM in the CPU/keyboard, don't bother. Since part of it is used for "housekeeping" and part of it contains this program and its variables, you can't really check all the locations, so any test like this would have no meaning. If necessary, you could swap those eight chips for eight from the Expansion Module, and run the test again. ■

Program Listing

```
100 CLS PRINTTAB(18)"*** TRS-80 MEMORY TEST ***":PRINT
110 'BY MILAN D. CHEPKO, M.D.    THIEF RIVER FALLS, MN  56701
120 'JULY 1979
130 DEFINT A-Z
140 PRINT"THIS PROGRAM WILL TEST THE INTEGRITY OF THE RAM
150 PRINT"CHIPS IN THE EXPANSION INTERFACE. DETECTING ANY
160 PRINT"STORAGE ERRORS AND LISTING THE DEFECTIVE CHIP ".PRINT
170 PRINT"BEFORE STARTING THE TEST, YOU MUST PROTECT THE BLOCK
180 PRINT"OF RAM BY ANSWERING '32767' AS THE 'MEMORY SIZE'
190 PRINT:PRINT"THE BOUNDARIES FOR THE EXPANSION RAM ARE:
200 PRINT"   BLOCK 1 = -32768 .          -16385
210 PRINT"   BLOCK 2 = -16384              -1
220 PRINT"(YOU CAN ONLY TEST LOCATIONS BETWEEN -32768 AND -1)
230 PRINT:PRINT INPUT"HIT 'ENTER' WHEN READY";A$
240 CLS:PRINT"ACTIVATE LINE PRINTER (IF AVAILABLE)":PRINT:PRINT
250 '*** DETERMINES MEMORY ADDRESSES TO BE TESTED ***
260 INPUT"LOWEST (DECIMAL) ADDRESS TO BE TESTED";A(1)
270 IF A(1)>-1 GOTO260
280 PRINT:INPUT"HIGHEST (DECIMAL) ADDRESS TO BE TESTED";A(2)
290 IF A(2)>-1 GOTO280
300 '*** CHECKS TO SEE IF LINE PRINTER AVAILABLE ***
310 IF PEEK(14312)=63 THEN POKE 16414,141:POKE 16415,5
320 '*** BEGINNING OF TEST ROUTINE ***
330 PRINT:PRINT"BEGINNING FIRST PART OF MEMORY TEST
340 X=170:Y=85:PRINT:GOSUB440
350 IF J=0 PRINT"NO ERRORS ENCOUNTERED
360 IF J>0 PRINT"ERRORS AS NOTED
370 PRINT" ":PRINT"PROCEEDING TO SECOND PART OF TEST
380 X=85 Y=170:PRINT GOSUB440
390 IF J=0 PRINT"NO ERRORS ENCOUNTERED
400 IF J>0 PRINT"ERRORS AS NOTED
410 PRINT" " PRINT"TEST OF MEMORY LOCATIONS ";A(1);"TO ";A(2);"COMPLETED
420 POKE 16414,88 POKE 16415,4 END
430 '*** SETS-UP ALTERNATING BIT PATTERN IN MEMORY ***
440 J=0 FOR I=A(1) TO A(2) STEP2
450 POKE I,X:POKE I+1,Y
460 NEXT I
470 '*** CHECKS FIRST BIT PATTERN ***
480 Z=X:FOR I=A(1) TO A(2) STEP2
490 B=PEEK(I):IF B<>Z GOSUB570
500 NEXT I
510 '*** CHECKS SECOND BIT PATTERN ***
520 Z=Y FOR I=A(1)+1 TO A(2) STEP2
530 B=PEEK(I) IF B<>Z GOSUB570
540 NEXT I
550 RETURN
560 '*** FINDS THE DEFECTIVE BIT ***
570 J=J+1:FOR K=0 TO 7
580 E=INT(B/2[K)-2*INT(B/2[(K+1))
590 D=INT(Z/2[K)-2*INT(Z/2[(K+1))
600 IF E=D GOTO650
610 PRINT"BIT ";K;"INCORRECT AT LOCATION ";I
620 '*** FINDS THE DEFECTIVE CHIP ***
630 IF I>-16385 PRINT"   EXPANSION CHIP 2";8-K;"DEFECTIVE":GOTO650
640 PRINT"   EXPANSION CHIP 2";16-K;"DEFECTIVE
650 NEXT : PRINT" " RETURN
```

Note: The bracket in lines 580 and 590 is actually ↑.

Techniques for algebraic equation solutions on your 80.

Equations

Allan S. Joffe W3KBM
1005 Twining Road
Dresher PA 19025

Sooner or later, you may, as I did, explore what your computer can do to solve algebraic equations. There are exotic methods and there are simple methods. I chose the latter approach.

First I had to determine what I could *not* do, that is, solve an equation for its imaginary roots. These are roots that involve the square root of a negative number as part of the root or the entire root. A sample equation with two imaginary roots is $X^2 + 25 = 0$.

How Many Solutions?

This still leaves quite a bit of territory open for exploration, which can start with an equation such as $X^2 + 2 = 0$. Initially, you must recognize that the highest numerical value of the exponent of the unknown (here it is 2) represents the number of roots the equation has. Thus an equation with X^3 would have three roots, and one where the highest exponent value is Y^6 would have six roots.

This does not tell you how many of the roots may be positive, negative or imaginary. All it tells you is how many roots should exist. The root (or a zero of the equation, as it is sometimes called) is any value that, when plugged back into the equation, will prove the equation true. For example, the root for the equation $X - 1 = 0$ would be $+1$ because if $+1$ is inserted in place of X, the equation is true.

What we are trying to do so far is to define the problem at hand. While it is easy to succumb to the temptation to initiate a computer problem by playing "kitten on the keys," less frustration is encountered by first "putting the brain in gear."

Positive or Negative

We can get at least one more set of guideposts by observing what 17th-century mathematician Rene Descartes had to say about what the roots of an equation might be. Consider the equation $2X^3 - 5X^2 - 4X + 3 = 0$. Descartes postulated that if you counted the sign changes of the terms of this equation, you would be able to predict the *maximum* number of positive roots that the equation might have.

Let's step through the equation term by term to see how this works. The sign of the first term is positive, the second term negative, the third term negative and the fourth term is positive. If we list these changes symbolically, $+ - - +$, we will see a series of two sign changes in total. This means that the equation can have a maximum of two positive roots. This does *not* mean that it *will* have two positive roots but that it can have no more than two positive roots. It can turn out that it will only have one positive root or it may have no positive roots.

Descartes then made available a simple way of determining the *maximum* number of negative roots that the equation might have. Consider the same equation but reverse the sign of any term having the unknown raised to an odd power. The sign order is now symbolically: $- - + +$, which shows one sign change.

This means that the equation can have a maximum of one negative root, with the same equivocation as for the positive root count—it might have no negative roots at all but will, in any case, not have more than one.

In applying this method for determining the possible number of negative roots, do not forget that if the variable appears without any exponent, it really is raised to the one power, so that any such term must have its sign changed to get an accurate count by this method.

The first step is to arrange your equation so that it is equal to zero. For example, if our equation had been in the form $2X^3 - 5X^2 = 4X - 3$, we would rewrite it in this form: $2X^3 - 5X^2 - 4X + 3 = 0$.

Since the TRS-80 demands that the equation be entered in such a way that it looks like this, $0 = 2X^3 - 5X^2 - 4X + 3$, we will do so or its electronic insides will give us an unwanted error message. At the same time we will, for the sake of elegance, use the letter Y in place of zero.

$Y = X^3 + 3X^2 - 10$	+1.49
$Y = X^3 - 3X - 1$	+1.88, −.347, −1.53
$Y = X^2 + X - 3$	−2.3, +1.3
$Y = X^2 + X - 6$	−3, +2
$Y = X^4 + 4X^3 - 6X^2 - 20X - 23$	−4.60, +2.60

Example 1.

The Program

If we have made it this far, it must be time for a program. It is, but every program has to work on some premise that we hope will be true and proper. Our implied premise is that if we find some value of X that, when inserted back into the equation, will make it equal to zero, then that value will be a root or zero of the equation; hence, at least part of the solution of the equation has multiple roots.

In essence, we are going to "guess" a number that we think is the solution and plug it into the equation to see if we get zero upon doing so. Since we have available a computer, whose strong point is repetitive calculation, we can insert our guess and have the computer either increment or decrement the guess in small steps which it can do rapidly.

Program A shows a simple program that uses the computer as a scorecard or a scratch pad. I call it the "Let your eyeballs do the walking through the Yellow Pages" approach.

When you hit RUN, the computer is ready for you to input X and D. X is your initial guess, and D is the increment or decrement factor you wish applied to X.

An excellent trial guess for X is 1, and a corresponding excellent value for D is 0.1. This information will speed up your use of the program. The only time the general choice of 1 for X will fall short is if 1 happens to be a root. You should also keep in mind that X and D can be present in various sign combinations:

X	D
+	+
−	−
+	−
−	+

all varieties of which may be useful in obtaining the roots of the more complex equations.

```
10 CLS
20 INPUT X,D
30 X=X+D
40 INSERT EQUATION IN THIS LINE
50 PRINT X,Y
60 GOTO 30
```

Program A.

Let us start with a simple equation to demonstrate the methodology. Insert the following equation into line 40: $Y = X^4 + 8X + 12$. In TRS-80 form this will be: `Y=X↑4+8*X-12`.

When we run the program, the ? appears indicating that we should input our X and D values, separated by a comma, on one line. As indicated we have to make a choice of sign for both X and D, so let's choose positive values for both. Your entry would be: `? 1.,1`.

Now hit RUN and let the screen fill with about ten or 15 lines of X, Y values and stop the display by using SHIFT @. Examine the listing and you will see the following sequence of lines.

1.1	−1.7359
1.2	−.3264
1.3	1.2561
1.4	3.0416
1.5	

The first column is X values; the second is Y values. Notice that the series of Y values converges toward zero and that there is a sign change between X values for X = 1.2 and X = 1.3.

A sign change in the Y column shows the value of X that will satisfy the equation. At this stage of the game, this is an approximate answer. To refine the answer, we take the value of X just before the sign change of Y and rerun the program using this value for X. To refine the precision of the answer, we fine-tune our D value, adding at least two more decimal places to D (if the initial root is less than 1, the next D value should be .01). Thus D was originally .1 and now becomes .001.

We rerun the program using the new X and D values and again scan the display using the SHIFT @ to control the display. This time when we detect the sign change, you will see opposite the newly determined value of X (1.221) that the value in the Y column reads −9.39179 E−03. Whenever you get a value such as this in the Y column (i.e., a value accompanied by a negative exponent), you can hang up your hat and call it a day. The computer has delivered a value for X that, if plugged back into the equation, will satisfy it to within a gnat's eyelash.

If you had examined the equation and applied Descartes' rule of sign changes, you would have determined that there is at least one positive root (which we have just found) and one negative root. Well, how do we get the negative root to come out of the woodwork?

Generally, when the positive root has been found with X and D of 1 and .1, you can usually find the negative root by applying the identical values but with the sign changed. Thus we run the program using X = −1 and D = −.1.

Now upon running the program you will get a series converging toward zero and showing an area where there is a sign change between two successive lines. When you spot the sign change, stop the display, remembering that the sign change we are looking for is only one that takes place in the Y column of figures.

The portion of the display where the sign change takes place is now listed:

−2	−12
−2.1	−9.3519
−2.2	−6.1744
−2.3	−2.4159
−2.4	1.97761

Thus our rough answer for the negative root value of X is −2.3, since it is opposite the last negative value in the Y column before the sign change.

We can now refine the negative root in the same manner as we did for the positive root. This time our new X is −2.3 and our new D is −.001. When the program is again run, you will spot opposite the value −2.357 in the X column the value 6.81877 E−03 in the Y column. Your calculation has given you a good value for the negative root of the equation.

A brief tabular listing of the values around the area of the answer shows:

−2.355	−.0818005
−2.356	−.0375195
−2.357	6.81877 E−03
−2.358	.0512257

If you need further accuracy, you can make the new X value −2.357 and the new D value −.00001 and run the program again. This time when your Y value with the negative exponent shows up, the absolute value of Y will be so close to zero that you'll have to split the gnat's eyelash.

As a clinical exercise this might be worth doing at least once, but for practical calculation, it is a poor use of the electrical juice. It is certain that good judgement is better than pushing the limits of technology just so you won't get that feeling you have somehow let the "machine" down.

Watching the Xs Go By

There are equations that will start with a series of Y values that diverge from zero; in other words, as you watch the screen, the Y values proceed to get larger. This is not always caused by an unfortunate sign choice for the X and D values, although this is the most likely cause.

There are equations where the generated Y values do indeed diverge from zero, but if you are patient and allow at least one screen's worth of values to go by, you are often rewarded by seeing the generated Y values start to diminish in value and finally generate a sign change in the Y column, which means you have found a root of the equation.

Let us run through one more equation to solidify the method. Consider $X^4 - 4X^3 - 6X^2 + 20X + 9 = 0$. According to the sign convention, it may have two positive roots and two negative roots.

If we first search using X = 1, D = .1, we will get the answer 2.4 at our first sign change point. If we then rerun the program using X = 2.4 and D = .001, we will get X = 2.414. Thus we have found one of the positive roots.

Next we can search for another possible root by using X = −1, D = −.1. The X value this search turns up opposite the first sign change in the Y column is −2.1. If we again refine this by rerunning the program using D = .001, we will come up with the answer for this negative root of −2.162.

Our next search will utilize one of the remaining two pairs of possible X and D sign possibilities; this time X = 1, D = −.1. Running this pair in the program will produce an initial X value of −.4, and refining this in the same manner as before will produce a final X value of −.4139.

With the experience you have gained from running this program for many equations with known roots, you should know that when you have succeeded in smoking out two possible roots of like sign (here we have the two possible negative roots) and one root of the opposite sign, it usually means that you should find the missing positive root (in this case) by returning to the original sign pair that produced the first positive root. This was X = 1, D = .1.

Since you have already used the pairing X = 1 and D = .1 to produce the first positive root, the next guess to insert is two times the previous value of X, which, in this case, is X = 2. If you run these values (keeping D = .1), you will see that a positive root is indicated, *but* it is the same value as the first positive root developed. Once again double the last value of X so that now X = 4, again keeping D as .1. Now when you run the program, a positive root of 4.1 will be produced. You can now use X and D values of X = 4.1 and D = .001 to refine the answer; the missing positive root is 4.162.

Following this general procedure, I have found that if the missing root is not found with a still further doubling of the X value, you may be rather certain that the "missing root" does not exist as a real root but is an imaginary root. That X in the equation is raised to the fourth power guarantees the existence of four roots but does not tell you how few or how many of them may be imaginary roots.

```
10  for X = −5 to 5 step .1
20  Enter equation in this line
30  Print x,y
40  Next X
```

Program B.

There exist many algebra books that will provide you with all sorts of theorems, postulates, etc., to enable you to prognosticate just how many roots are real and how many are imaginary. However, if you are in my ring and have not grappled with classical algebra for several decades, this could produce severe mental strain. I prefer the use of the computer plus a bit of educated guesswork.

The technique presented here is very simple, taking more words to describe than it takes time to implement. There may come a time in your life when you can't find a root of an equation, a root that blind intuition says does exist. As a fast check in such a situation, we can go from the simple to the barebones. Consider the four-line program in Program B.

This will produce a readout stream on your video display that will stop when the assigned limits of X have been reached. If you watch the display for sign changes in the Y column, just as with the previous efforts, you will see that the roots in the X column will be printed beside the last value of Y prior to any sign change in the Y column.

The values for X listed in line 10 of Program B are practical for any equation where the unknown is raised to the third power or higher. It is obvious that a term having the unknown raised to the highest power will usually be the determining factor in fixing the value of the sum of the parts of the equation. For an equation where the unknown is raised only as high as the second power, change line 10 to read: 10 For X = −9 to 9 step −1.

I hope this will ensure that you do not miss a possible root in this type of equation where the second power of the unknown might not be that powerful in fixing the composition of the roots. Remember, this whole method of solving equations depends on your using the ability you have to reason, coupled with the ability of the computer to carry out the results of that reasoning process in a rapid mechanical fashion.

I have included a series of equations with the known real roots (see Example 1). By running these samplers, you can gain familiarity with just how simple the techniques are for playing algebra through, what my wife has up to now called, "your kilobuck etch-a-sketch."■

If you intend investigating machine code, a hex keypad can make life easier.

Babybug Keypad

Dennis Bathory Kitsz
Roxbury, VT 05669

If you find machine-language programs enjoyable and challenging, you might be interested in performing a simple yet powerful modification to your TRS-80. Adding a hexadecimal keypad, numbers 0 to 9, A to F, plus backspace and enter, is inexpensive and easy.

The decimal numbers function normally, and all the characters can be used in conjunction with Babybug (Feb. *80 Microcomputing*) for quick keying of machine code.

Radio Shack offers its decimal-only keypad conversion for around $70; a complete hex keypad is available from Jameco Electronics (1021 Howard Avenue, San Carlos, CA 94070) for $10.95 plus shipping. Since it is not encoded, it is easy to parallel-connect its keys to the main keyboard and, by using one dead key, turn your TRS-80 into a powerful microcomputer.

I converted my computer in about two hours. All you need (other than a dose of warranty-voiding courage) is some wire, a soldering iron, two ten-inch long ½-inch by ½-inch pieces of plastic rod, five-minute epoxy, some assorted tools and a hot razor blade.

Photo 1. Keyboard and CPU board after removal of ROMs and mounting of hex keypad. Note cable to Level II ROMs at bottom center of photo. ROMs are remounted in the feet of the case bottom.

How to Start

First, carefully undo the cabinet (noting the different sizes of screws used to fasten it together). Take the electronics out of the case and set everything on a spacious work surface. You will notice that the Level II ROMs are fastened to the right of the keyboard with double-face tape; these will have to be lifted off the circuit card.

The interconnect cable to the ROMs is long enough to remount them inside the base of the cabinet, in one of the depressions that serve as feet. If the double-face tape has not been damaged, they may be fastened immediately; if the tape cannot be reused, fasten them by some other method, but be cautious not to cover them as heat buildup shortens their useful lives.

The Jameco Electronics keyboard base is identical to the TRS-80's in height and depth, so the two ten-inch plastic strips can serve as a rigid trailer hitch for the smaller board. Carefully support both boards so they are parallel and the hex pad meets the TRS-80 printed-circuit base. Cement the plastic strips in place; the vertical alignment of both sets of keys should now be identical.

When the glue has set, use the plastic shell to design a cardboard template of the current key positions and, with the aid of a ruler, draw extension lines across the right side of the template. These become the upper and lower limits of the new keypad opening. Align the template with the new double keyboard assembly and mark the vertical positions of the keys, allowing about 1/32" additional on both sides. This should bring you within ¼ inch of the pilot LED.

Remove the black portion of the cover by turning it over and snapping it out of the six locking tabs along the edge. Using the template and a hot razor blade, carefully cut an opening, working from the face of the soft plastic cover.

This procedure is time-consuming and must be done with care to achieve an as-built appearance. Double check all work to make sure the cover fits over both keyboards and that the keys are free to move (trim or file where necessary). If you have made any nicks in the soft plastic face, rub them with a glass marble and they should disappear into the background of the face texture.

Interconnect the Wires

Now it is necessary to interconnect wires from the hex keyboard to the main keyboard's printed circuit card. Rest the keyboard on its face but take care not to strain the band of wires connecting the card and CPU board. Set aside the white plastic spacers, and keep the board well supported during soldering.

For interconnections, I used wire-wrap wire because it was thin and flexible, although any

fine wire will do. First, solder all the connections at the hex pad as shown in Fig. 1. You will be soldering to very fine terminals which are part of the key contacts and springs, so be careful to avoid excessive solder or heat. The solder should flow easily onto the gold-clad contacts.

Next route individual wires from the hex key groups to the points printed on the circuit card, as indicated in Fig. 1. If you have a TRS-80 technical manual, you may notice that some of the keys do not match the wiring on the master schematic. I have a very early TRS-80 and have not been able to check this difference with other Level II machines, so I recommend following the circuit traces to assure that your parallel wiring matches that of the keys pressed.

Once you are sure all the connections are properly made, flip the board over. Avoid putting stress on the keyboard interconnect cable. Replace the white plastic spacers, insert the cables to the monitor and power supply and power-up the computer.

Type all the characters on the main keyboard to make sure it operates normally, then type the letters and numbers on the hex keyboard. Also test the backspace (at this point the bottom right-hand key is dead). Press the button currently marked SHIFT (this will be the enter key), and the lines of randomly typed test characters should certainly cause a ?SN ERROR message.

All keys should now be working properly. Any problems will occur in the form of incorrect letters or patterns of repetitive letters caused by incorrect or shorted wiring of the new keypad. If most characters work, but some do not, a wiring error is likely.

Recheck all wiring, and if no problem is evident, insure that the wiring shown in this article is correct for your machine. (Master schematic differences in my unit involved the backspace and enter keys.) Check also for solder splashes or damaged printed circuit board traces.

Remove the cables and power, gently reinsert the circuit cards into place and replace the white spacers, the keyboard assembly and the cover. Take care that the LED power light sits in the front panel and that all the keys move freely. After replacing the screws and cables you are ready to run Babybug again.

Try the programs described in Part I of this article. The new keyboard should speed the process along considerably.

The Control Key

The absence of specialized function keys, on the TRS-80, confines the user to type-written commands. Its simplicity makes

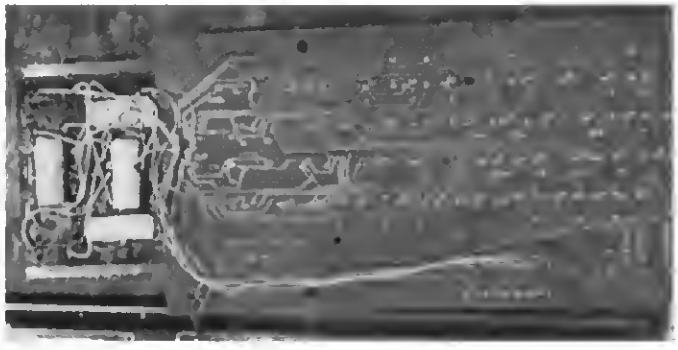


Photo 2. Wires run from points on the main keyboard's p.c. card directly to hex keyboard. White rectangles are plastic foam that helps cushion the author's abusive typing style.

the TRS-80 a very accessible machine, but with that accessibility have come a few disadvantages. Sometimes the need arises for an escape from a program without BREAKing in, or for some additional control over a program's execution. The dead key on the new hex keyped offers those special functions.

This control key (dead key) is valuable when using your own machine-language modules. This key can send the contents of the screen to the line printer; convert to lowercase*; direct an array of edit commands in a word-processing program; call up any number of machine-language modules; or obtain TRS-80 graphics from the keyboard.

Let's first see how this key might be used in BASIC. Like the shift key, the control does not produce a character by itself, but changes how the computer reads another character typed simultaneously. Set up and RUN the following BASIC program:

```
 5 CLS
10 A = PEEK(14464)
20 IF A = 128 GOTO 50
30 A$ = INKEY$ : PRINT A$;
40 GOTO 10
50 A$ = INKEY$
60 IF A$ = "" GOTO 10
70 PRINT CHR$(ASC(A$) + 101);
80 GOTO 10
```

As you type, letters will appear on the screen as they would on the page. With the exception of some of the command keys like ENTER everything operates as before.

Now shift with the new control key and type. The screen displays TRS-80 graphics! Release the control key and normal typing resumes.

How does this work? Each key occupies a position in a grid

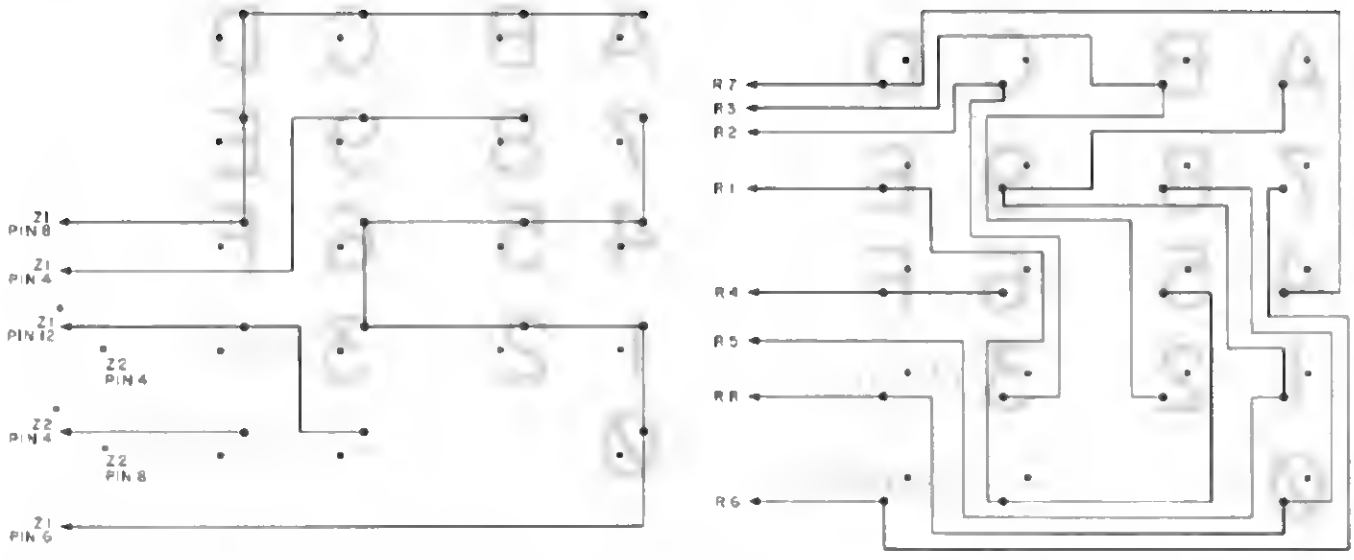


Fig. 1. Wiring of (A) address and (B) data lines on the hex keypad. Lines marked with an asterisk differ from the pin connections on the TRS-80 master schematic.

Address	PROGRAM	Description	Mnemonic
4C00	F5	Save the current contents of the accumulator and the condition flags	PUSH AF
4C01	3A 80 38	Load the accumulator with the contents of memory location 3880 (where SHIFT and CONTROL are located)	LD A,(3880)
4C04	FE 80	Compare the contents of the accumulator with 80 hex (which is the number produced when CONTROL is depressed). If it compares exactly, set the "zero" flag.	CP 80
4C09	28 04	If the zero flag is set (meaning here that the CONTROL key was depressed), jump ahead four places.	JP Z,04
		The next two instructions would be executed ONLY if they were not jumped over by the previous instruction:	
4C08	F1	Reload the accumulator and flags with their original contents before this "patch" program.	POP AF
4C09	C3 58 04	Jump to 0458, the place where the keyboard scan "switchboard" originally directed.	JP 0458
4C0C	3A 01 38	Load the accumulator with the contents of memory address 3801, the address of keys @, A, B, C, D, E, F and G.	LD A,(3801)
4C0F	FE 02	Compare the contents of the accumulator with 2, the number produced if letter "A" is depressed	CP 02
4C11	20 EE	If the result does not compare exactly (i.e., the "zero" flag is not set), jump back 18 steps to address 4C01; if it does compare precisely . . .	JP NZ,EE
4C13	C3 00 00	. . . jump to 0000, the beginning of BASIC	JP 0000

Program Listing 1.

wired so the processor interprets it as a memory bank—one line of addresses crossing one line of data. The keyboard is scanned, and a depressed key drops a one into its allotted data bit in a memory location. The keyboard-scanning subroutine in BASIC checks the memory locations in order, and, upon finding a one, jumps to another routine that displays that key's character upon the screen.

There are many additional routines to keep keys from endlessly repeating, to differentiate between character keys and command keys and to keep up with fast typists.

The memory slot for our new key is 14464 (3880H). When the shift key is depressed, a one appears in this slot's least significant bit (00000001); the control key is wired to insert a one in the most significant position (10000000).

BASIC's keyboard scanning routine is written to read the shift key, but is blind to the new control key. The only way to find whether the key is pressed is to PEEK into location 14464 (line 10 of the program).

The result of that PEEK would be either 1 (SHIFT) or 128 (CONTROL). Our BASIC program ignores the SHIFT, displays whatever key is depressed and loops back to the beginning of the program. If it finds the control key depressed, it goes to line 50, searching (using INKEY$) for another key to be depressed.

It continues with the above loops until it finds both the control key and a character key depressed. Determining the letter's ASCII value, the program adds 101 (to alter the value to that of a graphics character) and displays the result. The new control key makes this a simple way to display both characters and graphics.

A Telephone Switchboard

To exploit the broad powers of this control key, it is necessary to know a few things about the TRS-80 design. The ROM (Read-Only-Memory) is fixed, and contains the BASIC language. It occupies about 12,000 bytes of space, with some unused areas set aside for future improvements.

Upon power-up, the ROM executes a great deal of housekeeping, the most important part of which is cordoning off a section of volatile memory for storing temporary information and establishing a kind of "telephone switchboard" located in the area of memory about 1,000 bytes long, starting at location 4000H.

Many of the BASIC subroutines momentarily jump to this area, and then back into the BASIC ROM, to a location set up during housekeeping. The exciting aspect of this switchboard is that we can patch our own call through it! The control key will use one of those patches.

At the start of BASIC's keyboard scan routine, it jumps to location 401D in the switchboard to receive instructions for its next move; housekeeping has inserted 0458 into locations 401E and 401F. We will be changing this jump, but before we modify anything in the complex BASIC language, it will be necessary to decide precisely what we want to do with the control key.

My first use of the control key was to gain a new command: Return to MEMORY SIZE?, without turning off the computer and losing the contents of memory or wasting my only USR(0) location. Program Listing 1 checks to see if the control key and letter "A" are depressed. If both conditions are met (in that order, as with the SHIFT key), it returns to a MEMORY SIZE? condition (prepare the program with the aid of Babybug).

After you have prepared the program, your next move is to patch it into the keyboard scan. Since the keyboard's scanning

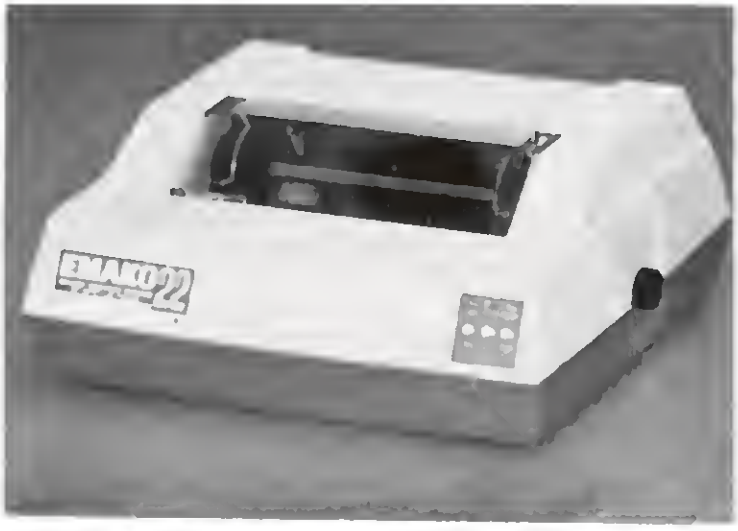
	05	DEC B
	B1	OR C
	B5	OR L
Results of decrement in	ED B0	LDIR
Results of decrement in	10 Nn	DJNZ 00

Table 1. The ZERO flag is effected by (among others less important for these few simple programs) these instructions.

Lords Corp

presents

THE PERSONAL PRINTER for TRS-80



FACT: The Super Brain (Emako 22) printer is one of the most co$t effective, feature packed printers available. The Emako 22 is Bi-Directional and has a 9x7 dot-matrix print head, 9.5" adjustable width pin-feed carriage. Prints a 132 col/line. Printing selectable: (132 or 80 ch/line normal, 40 or 66 ch/line double width). 100 million char. print life can be expected. 96 ASCII characters means lower case is available especially for both of Radio Shack's TRS-80 Mod 1 & 2.

Now you can get your own hardcopy only $728. Cables $38.
Bank Cards Welcome.

LORDS CORP ✓152
P.O. Box 99
Port Angeles, WA 98362
Telephone (206) 457-3064

QUALITY TRS-80 SOFTWARE

KEYWORD Indexing System

A series of programs that will create a data file on disc, build an index of all occurrences of "keywords" in the text of the data file and allow inquiries or searches into the file using the indexed keywords. The system features
- flexible record lengths with location pointers
- deletion of non keywords from index by system
- "and" "or" "not" logic for inquiries
- interface for user written inquiries

KEYWORD INDEX—2 disc 32K DOS system $39.95

SORTS for HOME and BUSINESS

No computer user should be without a versatile, easy to use sort program. The Northeast Microware in memory sort programs are written in Level II BASIC and have the following features
- Sort ALPHA or NUMERIC data
- Sort on up to 5 fields simultaneously
- In ascending or descending sequence
- Supports kb, video or tape I/O
- Supports seq. disk and printer I/O (SORT HD)
- Supports user I/O routines
- User exits (SORT HD only)

SORT II—16K Level II in memory sort $19.95
SORT HD—32K DOS in memory sort $29.95

FOR the SERIOUS GAMBLER

BLACKJACK SIMULATOR. Allows you to simulate the playing of thousands of hands of BJ and analyze the results on tape in Level II BASIC. $19.95

Manuals for all programs available for $3.00 ea.
(price deductible on purchase of program)

"TRS-80 is a registered trademark of TANDY CORP."

Northeast MICROWARE
BOX 6153, ✓74
SYRACUSE, N.Y. 13217

AT LAST

USEFUL HOME PROGRAMS For Your Level II 16K

CALCULATOR PLUS turns your TRS-80 into a printing calculator! Excellent as an on-screen calculator without a printer. All functions double precision, chain and mixed calculations, 5 memories with titles—can be used as constants, item count in add and subtract with on-screen or printed review of entries, optional dollar format. Prints all calculations, results and memories, ideal for use with Quick Printers. Loads fast, easy to use.
MPC-1 $9.95

CHECKBOOK PLUS solves the problem of monthly bank-statement reconciliation. No cumbersome tape record keeping. Just do your checkbook once a month, and let Checkbook Plus handle all the details and find the errors.
MCB-1 $9.95

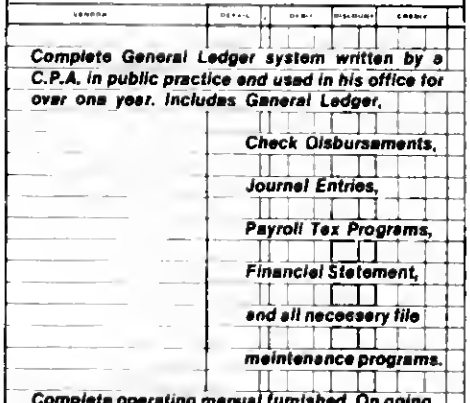
CALCULATOR PLUS and **CHECKBOOK PLUS** both on one cassette.
MC-2 only $14.95

Each cassette includes a bonus Loan Payment program with clean dollar-format on-screen monthly or yearly display of payment, interest, capital and balance.

"TRS-80 is a registered trademark of TANDY CORP."

MANHATTAN SOFTWARE, Inc.
P.O. Box 5200 Grand Central Station
New York City, New York 10017
✓90

TRS-80
client write-up system



Jerry E. Bartram, CPA
25455 Barton Rd. B-209
Loma Linda, CA 92354
(714) 825-2736. ✓137

program cannot be modified from the keyboard, write the following line:

```
100  POKE 16414,00 : POKE 16415,76
```

This is the address 4C00H broken into two decimal pieces; when you RUN 100, the patch will be inserted. At first you will see no difference, and all keys should function normally; the patch is totally transparent to BASIC, to programs and to all keyboard functions. Unless the processor finds the control key depressed, it skips right back to the scanning routine (the control key too will have no effect by itself).

Now depress control plus "A". You will be returned instantly to MEMORY SIZE? This control-plus-letter concept is formidable; each key on the keyboard can represent the pathway to an entire program, that is, it can represent a command independent of BASIC!

Challenge Your Skills

As a challenge to your new programming skills and to protect some memory, reload Babybug to create a program that performs the following:

1. The program is transparent to BASIC.
2. The program uses the control plus letter "A" to call up the white-screen module.
3. The program uses the control plus letter "B" to call up the cassette-load module, but loads only 255 bytes, then returns to BASIC.
4. The program uses the control plus letter "C" to call the "BASIC Bounce" module, listens to and displays only 255 bytes, then returns to BASIC.

To write this program, you will need some additional information and a few hints.

First, do not let your cassette-load module overlay its input data atop any of your current programs. Place it well out of the way, in your highest memory area. Remember to POP information off the stack as many times as you PUSH it on and note that every time you call a subroutine, the program's current address is PUSHed onto the stack.

Letters A, B and C are in memory location 3801H. A is data 02 (binary 00000010), B is 04 (binary 00000100) and C is 08 (binary 00001000).

If you need more loops than you have registers, remember that the current value of a loop can be stored in memory while you perform other operations and retrieved when it is needed. For example, if you start a loop in the B register, you can store it and use the B register for another function.

Later, you can store that information and retrieve your loop status; decrement the loop's value, store it; and again you can retrieve the stored information.

The author wishes to express his thanks to Philip K. Hooper for his encouragement, and for his astute criticism of the software content of this article, as well as to Stan Ockers for his improvements to Babybug for non-typists. ■

Notes to Text

*An upper/lowercase character-generator circuit is already part of the TRS-80, but no access has been provided. The Peripheral People (P.O. Box 524, Mercer Island, WA 98040) offer free, excellent lowercase conversion information. Both hardware and software instructions are included, and the conversion can be achieved with just a few ICs.

Call a subroutine	CD Lx Mx		CALL 0000
Call a subroutine if a zero flag is set	CC Lx Mx		CALL Z,0000
Call a subroutine if a zero flag is not set	C4 Lx Mx		CALL NZ,0000
Compare the contents of the accumulator with an integer	FE Nn		CP 00
Decrement the B register	05		DEC B
Decrement the C register	0D		DEC C
Decrement the HL register pair	2B		DEC HL
Decrement the DE register pair	1B		DEC DE
Decrement the BC register pair	0B		DEC BC
Increment the B register	04		INC B
Increment the C register	0C		INC C
Increment the HL register pair	23		INC HL
Increment the DE register pair	13		INC DE
Jump to memory location	C3 Lx Mx		JP 0000
Jump to memory location if a zero flag is set	CA Lx Mx		JP Z,0000
Jump to memory location if a zero flag is not set	C2 Lx Mx		JP NZ,0000
Jump a relative distance	18 Nn		JR 00
Jump a relative distance if a zero flag is set	28 Nn		JR Z,00
Jump a relative distance if zero flag is not set	20 Nn		JR NZ,00
Load the accumulator with an integer	3E Nn		LD A,00
Load the accumulator with the B register	78		LD A,B
Load the accumulator with the H register	7C		LD A,H
Load the accumulator with contents of memory location HL	7E		LD A,(HL)
Load the accumulator with contents of memory locatino DE	1A		LD A,(DE)
Load the accumulator with contents of memory location	3A Lx Mx		LD A,(0000)
Load the contents of memory location HL with the accumulator	77		LD (HL),A
Load the contents of memory location DE with the accumulator	12		LD (DE),A
Load the contents of the HL memory location with a number	36 Nn		LD (HL),00
Load the HL register with two bytes	21 Lx Mx		LD HL,0000
Load the DE register with two bytes	11 Lx Mx		LD DE,0000
Load the BC register with two bytes	01 Lx Mx		LD BC,0000
Load the B register with a byte	06 Nn		LD B,00
Load the C register with a byte	0E Nn		LD C,00
Load the B register with the byte found at HL memory location	46		LD B,(HL)
OR the contents of the accumulator with the C register	B1		OR C
OR the contents of the accumulator with the L register	B5		OR L
PUSH the two bytes in the HL register onto the stack	E5		PUSH HL
PUSH the two bytes in the AF register onto the stack	F5		PUSH AF
PUSH the two bytes in the BC register onto the stack	C5		PUSH BC
POP the top two bytes in the stack into the HL register	E1		POP HL
POP the top two bytes in the stack into the AF register	F1		POP AF
POP the top two bytes in the stack into the BC register	C1		POP BC
Load the memory contents of HL into the memory location DE; increment HL and DE; decrement BC; loop back if BC not zero	ED B0		LDIR
Decrement BC; loop back indicated distance if not zero	10 Nn		DJNZ 00
Return	C9		RET
Return if the zero flag is set	C8		RET Z
Return if the zero flag is not set	C0		RET NZ

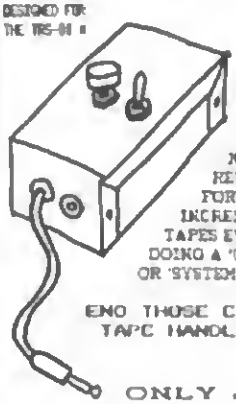
(Lx = Least Significant Byte, Mx = Most Significant Byte, Nn = One-Byte integer)

Table 2. These instructions are a limited selection of the Z-80 instruction set; there are nearly 700 commands in total though the program can be effectively created with just these.

Photo 3. Finished modifications on the TRS-80 before cover is reinstalled. Note the four-conductor ROM cable that runs over top of CPU board, and the author's lowercase modification switch at bottom.

From The Engineering Group At GSIC!

DESIGNED FOR THE TRS-80 ®



CASSETTE REMINDER INTERFACE

NOW YOU CAN REMIND, FAST-FORWARD, OR INCREMENT YOUR TAPES EVEN WHEN NOT DOING A 'CLOAD', 'CSAVE' OR 'SYSTEM' COMMAND.

END THOSE CASSETTE TAPE HANDLING BLUES!

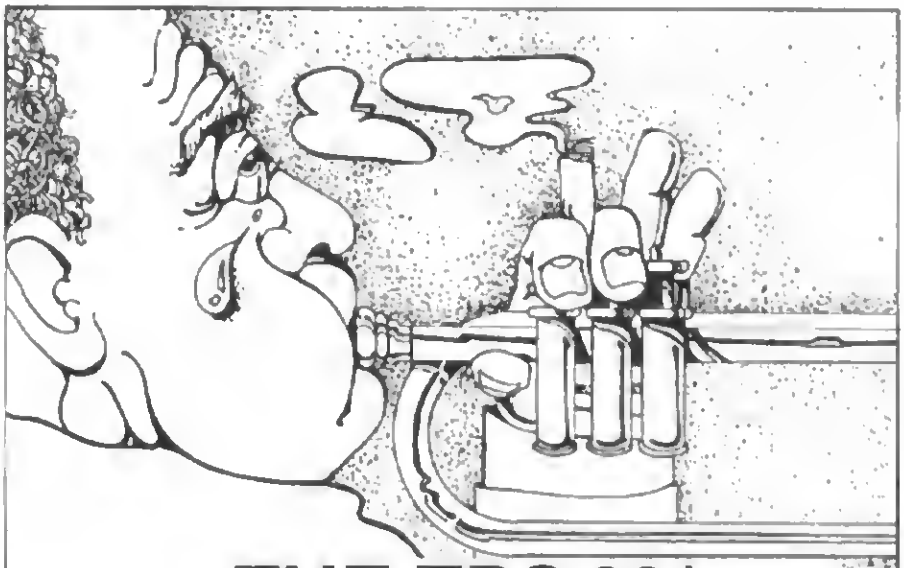
ONLY .. $14.95

WE PAY ALL SHIPPING AND HANDLING CHARGES.

FEATURES: Adapts instantly to present system, precision toggle and momentary push-button switches, durable painted aluminum cabinet, high quality micro-plug and jack, non-slip rubber feet, does not interfere with normal system operation, compact design complements your system.

SEND CHECK OR MONEY ORDER TO: ✓93
GRANITE STATE INSTRUMENT COMPANY
Box 3486
Nashua, NH 03061

(® TRADEMARK OF TANDY CORPORATION)



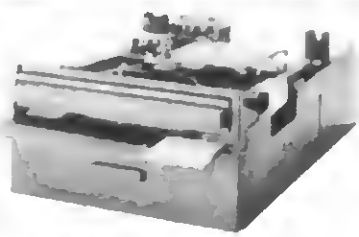
THE TRS-80* BLUES.

Do your TRS 80'S limitations hit a sour note? Our full line of business-oriented software can change that tune. They're easy to use, modularly designed, and superiorly documented, giving you versatility and sophistication of a higher scale, at a price you'll sing about.
So write us, and we'll send you the score.

SMALL BUSINESS SYSTEMS GROUP
GROTON STREET, DUNSTABLE, MA. 01827 (617) 649-9595 ✓18
*TRS-80 is a registered trade mark of Radio Shack, a division of Tandy Corp

Sirius Systems introduces lower prices to quality drives!

Remex RFD 4000/8" Floppy Disc Drive
Double the storage!
Double sided . . .
Double density!!

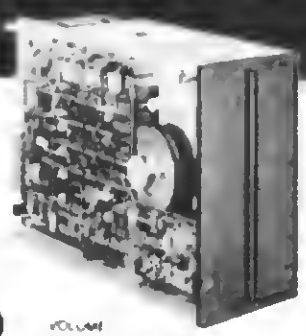


$525

Offers quality and features found in drives costing much more! ■ Single or Double Density ■ Double-Sided Drive ■ Door Lock INCLUDED ■ Write-Protect INCLUDED ■ 180 Day Warranty ■ Compatible with Shugart 850/851 ■ Low Power Operation ensures LONGER LIFE!! ■ Model RFD 4001 offers Data and Sector Separator

AVAILABLE OPTIONS/ACCESSORIES
- ☐ Dual Drive Power Supply and Cabinet $139.95
- ☐ Single Drive Power Supply and Cabinet $119.95
- ☐ Interface Manual, $2.95
- ☐ RFD 4000 Manual $5.95
- ☐ Drive Cabinet $29.95

Remex 10008
If you've been looking for a less expensive floppy disc drive, but not wanting to sacrifice quality— your search is over!



$395

VOLUME DISCOUNTS AVAILABLE

You get both in the Remex 10008! For only $395 look at what you get ■ 8" Floppy Drive ■ Single or Double Density ■ Hard or Soft Sectoring ■ Media Protection Feature ■ Single Density Data Separator ■ 180 Day Factory Warranty

AVAILABLE OPTIONS/ACCESSORIES
- ☐ Door Lock $19.95
- ☐ Write Protect $19.95
- ☐ Connectors $9.95
- ☐ Dual Drive Power Supply $91.95
- ☐ Single Drive Power Supply $69.95
- ☐ Drive Cabinet $24.95
- ☐ Interface Manual $2.95
- ☐ Interface Adapter $12.95 Remex to Shugart

SIRIUS 80+

The Perfect Add-On for your TRS-80*

- Comes complete ready to plug in and run!
- Same track to track

SIRIUS 80 • 1 (Single Head) **$349.95**
SIRIUS 80 • 2 (Dual Head) **$419.95**
*TRS-80 is a Tandy Corp.

MPI-51/52

A Great Reliable Mini-Drive!

- Fast! 5ms track to track access
- Exclusive Puffer Band Design
- Unique Doorlever Mechanism
- Reliable 1½% Speed Stability

MPI-51 (Single Head) **$259.95**
MPI-52 (Dual Head) **$349.95**

SIRIUS SYSTEMS ✓67
P.O. Box 9748 Knoxville, TN 37920
Phone Orders accepted 9AM-7PM (E.S.T.) 615-577-1072
☐ Check ☐ Money Order ☐ C.O.D. MC ☐ VISA ☐ AE

NAME _____ CARD # _____
ADDRESS _____ EXPIRATION DATE _____
CITY _____ STATE _____ ZIP _____ CARDHOLDER'S SIGNATURE _____

Add $7.00 per Drive for Shipping/Handling. Tennessee residents add 6% sales tax. Foreign orders add 10% (payment in U.S. currency only)

Why write software when you can use the inbuilt Level II assembly language subroutine!

Inside the ROMs

Bruce E. Stock
537 33rd Ave. South
Seattle WA 98144

One of the more fascinating aspects of owning a Level II TRS-80 computer is knowing that buried within it are over twelve thousand bytes of lovely machine-language routines just waiting to ease the load of the intrepid assembly language programmer. This fascination has caused me to spend hours poking about in listings, following chains of subroutines all over memory, to find some of the more immediately useful sections for my own assembly language programs. The search has been in turn frustrating, instructive, and rewarding. In this article I hope to share the rewards, while keeping the frustrations to a bare minimum.

Keyboard Routines

The keyboard routines are a good starting point. Let's begin with a top level routine for inputting data into a program. This routine is one step removed from the INPUT command in BASIC. The entry point of the routine is at 1BB3H (all addresses are given in hexadecimal). A call to this location results in a question mark on the screen, followed by a space. The operator is now free to input data, which is placed into the BASIC input buffer. As it's typed, the data also appears on the screen and you can backspace to correct errors.

Terminate the input by hitting the entry key. When the routine returns, the HL register pair contains one less than the buffer start address.

Now that the data is in the buffer, how do you get at it? Executing an RST 10H (Restart 10H) instruction advances the HL pointer to the first character entered (skipping any spaces), loads the character into the A register and sets the carry flag to indicate whether the character is numeric (C = 1) or alphabetic (C = 0). Each RST 10H executed loads the next character into A (again, skipping any spaces) for you to process. When the RST 10H returns a value of zero, you have reached the end of the entered data. If you don't want the question mark prompt printed at the beginning of the routine, enter it at 361H instead of 1BB3H.

Suppose you don't want a whole buffer full of data, but only a single keystroke (i.e., the INKEY$ type function)? A call to 049H returns when a key has been hit. The A register contains the character.

In many cases, you will want a routine which returns immediately after checking the keyboard, even if no key is punched. In this case, call 02BH. This routine will return immediately with A = 0 if no key is pressed at the instant the routine is called, or with A equal to the correct key code if a key is depressed. Neither of these routines displays the character entered. If the character must be displayed, use one of the routines in the following section.

The last keyboard routine to be discussed takes a decimal number in from the keyboard and returns with the equivalent binary value in the DE register pair. This is a very handy routine when the input needed is a decimal integer between 1 and 65,536. The routine used follows:

ENTRY POINT	DESCRIPTION	REMARKS
1BB3H	Prints "?", inputs data until enter is hit; displays data on screen.	Data goes into BASIC input buffer. 241 char max. HL points to location prior to first character on return. Uses AF, HL, DE.
361H	Same as 1BB3H, less prompt.	As above.
049H	Returns when a single character has been entered.	Character in A. Uses AF, DE.
02BH	Instantaneous read of keyboard	A = 0 on return if key not pressed, else A = character code. Uses AF, DE.
010H	Restart 10H. Advances HL, loads A with (HL), sets carry flag.	C = 0 if alpha, C = 1 is numeric. Uses AF, HL.
1E5AH	Processes decimal number into binary.	See text.

Table 1. Keyboard Routine Summary

ADDRESS	DATA	MNEMONIC	COMMENT
5000	21 50 50	LD HL,5050H	; load pointer into HL
5003	CD A7 28	CALL 28A7H	; output message
5006	CD B3 1B	CALL 1BB3H	; get input
		(process input)	
		End	
5050	59 4F 55 52	"YOUR	; message
5054	20 4D 4F 56	MOVE"	
5058	45 0D 00	CR	

Table 2. An example of the use of the message routine.

```
CALL  1BB3H
RST   10H
CALL  1E5AH
```

The first instruction gets the data into the input buffer. The second advances HL to point to the first character that was entered. The last processes all entered digits into a binary value which is left in DE. The three routines taken together use the A, DE, and HL registers.

Table 1 summarizes the keyboard routines. The Remarks indicate which registers are utilized by the called routine. You should save these registers if your program is also using them.

Screen Routines

The next series of routines deals with getting data onto the screen. We begin at the top again with a routine that prints a whole message on screen. Level II uses this routine to print the MEMORY SIZE message, as well as several others. To use this routine you must have a message stored somewhere in memory.

The message is a string of ASCII characters and must terminate with a byte containing zero. It may also contain control codes or graphics codes, but no quotation marks. First, the HL register must be loaded with the address of the first character of the message. Then a call to 28A7H does the rest.

Table 2 shows how it is done. In the example, the message "YOUR MOVE" is being output to prompt an input to a chess program. The processing routine following the 1BB3H call decodes the move input. The power of this call should not be overlooked, since, by including carriage returns, tab codes and so forth in the message, it is possible to format a complete display with a single call.

Now, let's display a single character at a time. Suppose that 049H has been called to get a character in from the keyboard and now it must be displayed on the screen. Call 32AH. This routine will take whatever is in the A register and print it on the screen without disturbing data in any of the other registers. The routine at 033H does almost the same thing, but it disturbs the DE register pair. These routines are good for many functions, for example, to shift to thirty-two (32) characters per line, just load 17H into the A register and call 32AH.

The last screen routine is short and sweet. A call to 1C9H clears the screen and sends home the cursor. The cursor may be turned on or off by loading 0EH or 0FH, respectively, into A, followed by a call to 32AH. The screen routines are summarized in Table 3.

Cassette Routines

Since Level II has a nice complement of cassette-related routines (CSAVE, CLOAD, INPUT #, etc.), it may seem unnecessary to go into more here. However, the routines described here allow input and output formats to best suit the situation. They can be used to improve the efficiency of data storage, or, as I have done, they may generate tapes that can be read by the BASIC SYSTEM command.

First turn on the cassette motor and record the synchronization pattern. This is done with a call to 284H. The synchronization pattern consists of 256 zeros, followed by a single byte containing the value A5H. Now that the motor is running and the sync pattern is on tape, record some data. Load the byte to be recorded into the A register and call 264H. When the final byte of data has been recorded, a call to 1F8H turns off the cassette motor. Be wary of doing too much data processing between output of data bytes, since the delays introduced could foul up the synchronization when the tape is read back.

To read in date which has been written on tape requires a call to 293H. This routine will turn on the cassette motor, read the leader until the A5H sync byte is found, print two asterisks in the upper right corner of the screen and then return. The following data can now be read a byte at a time by repeated calls to 235H. Again, don't spend too much time processing bytes between calls to 235H, or you may lose synchronization. When all data has been input, a call to 1F8H will turn the motor off.

If the idea of blinking the right asterisk appeals to you, each call to 22CH will reverse its state, on to off, or off to on. (The CLOAD and SYSTEM commands blink the asterisk each time a BASIC statement is read in or each time the checksum value is verified.)

Note that all the motor control routines described above are intended for use with the cassette plugged into the keyboard, not into the expansion box.

Table 4 summarizes the cassette routines.

Conclusion

The use of these routines should considerably simplify the I/O sections of your next program. Bear in mind, however, that all this does not come totally free. Most of these routines use pointers and buffers which reside within the area of RAM dedicated to BASIC. As a result, you can't locate programs in addresses below approximately 4300H, or you will disrupt the pointers. Also, several of the routines have additional features beyond those described, so, if you deviate significantly from the examples given, the results may be confusing, to say the least. Lastly, since the BASIC stack is used, the MEMORY SIZE cannot be set too low in value, or insufficient stack space will result.

These few restrictions are a small price to pay for I/O that is nearly as easy as using PRINT, INPUT, and INKEY$. With these routines available, the only real work remaining is to decide which of the programs you've been putting off will be the first you'll write incorporating them. ■

ENTRY POINT	DESCRIPTION	REMARKS
28A7H	Generalized message output routine	HL must point to message. Message must end with 00. Uses AF, BC, DE, HL.
32AH	Put one character on scrsen	Enter with character in A. Uses AF
033H	Put one character on screen.	Enter with character in A. Uses AF, DE.
1C9H	Clear screen.	Also sends home cursor to upper left corner

Table 3. Summary of Screen Routines.

ENTRY POINT	DESCRIPTION	REMARKS
284H	Turn on motor, write leader.	Uses AF. Leader is 256 zeros, followed by A5H
264H	Writes byte to tape.	Byte must be in A
1F8H	Turn off motor.	
293H	Reads leader and locates sync byte	Turns motor on. Sync = A5H. Uses AF
235H	Reads byte from tape	Returns with byte in A. Uses AF
22CH	Blinks rightmost asterisk.	Uses AF

Table 4. Summary of Cassette Routines.

A real work, real time application project for handy 80 owners.

A Home Brew Interface

C.R. Vince
27 Ventnor Way
Ottawa, Ontario
Canada K2J1M2

After becoming the proud owner of a TRS-80 system in April of 1978, I soon realized that the Level I, while an excellent teaching aid of the BASIC language, left much to be desired when it came to making my computer more than just an expensive toy. After waiting anxiously for several months (due, I suppose, to the extremely heavy demand), my Level II arrived and was installed.

Now I could really make my "toy" earn its keep, or could I? Yes I could, providing I put out another $439 (Canadian) for an expansion interface. But wait, all that would give me would be a real-time clock, mini-disk controllers, cassette and line printer controllers and "space for an additional PC board" (to add whatever), according to Radio Shack advertising. What about my home climate control, model railway control and other applications?

There had to be another way, and I hope that after reading this article you will agree with me that there is another way, perhaps even a better way, at least for hobby use.

Introduction

This article will describe an interface unit for the TRS-80 Level II that will provide the following features:

1) An interface board to the TRS-80 itself.
2) An output board having up to 16 8-bit parallel output ports.
3) An input board having up to 16 8-bit parallel input ports.
4) A TTY interface board.
5) A home climate control system.
6) A model railroad speed control system.

I would like to point out here that I am no expert in electronic circuit design. In this article, most of the circuits have been previously described in other books and publications, including *Microcomputing*. I have merely put them together in one package as simply and as economically as possible. However, the circuits have been tested and do work; in fact, they are in daily use.

To enable novices to understand the workings of the interface unit I have arrowed pertinent lines in the figures to show the directional flow of data on that particular line.

The unit has been built on five separate PC boards (excluding the power supply). I use the term PC boards loosely, as these boards were handmade, and only the common lines such as data bus, address bus and power lines were etched; other lines such as enable lines were wired.

Edge card connectors used were of the 62-pin type since they were the least expensive and most available at the time; consequently, pin connections given are for these 62-pin variety. Others such as 44 pin could be used providing they have enough pins to accommodate all lines entering or leaving the board. The edge card connectors were mounted on a piece of wood and like-numbered pins for the +5 V, ground supplies, the data and the address lines were multiplied from one connector to the next.

I strongly recommend the use of sockets for all ICs, as troubleshooting is made so much easier if you can simply replace a suspect IC to localize the problem. To emphasize this point, when I first plugged in the interface board, I had problems on a new IC, which seemed to work OK on static bench tests but failed in the unit. By simply exchanging two ICs, the problem was localized in minutes. In addition, don't forget to use .01 uF bypass capacitors on about every fifth IC.

Interface Board

To allow for expansion, I decided to use 74LS367s to buffer all signals coming from the TRS-80 (see Fig. 1). I initially buffered the data bus in both directions, but found that this caused problems because the interface

unit bus is, in effect, parallel with the internal TRS-80 bus, and each time an input to the Z-80 processor is effected (e.g., from memory), it also inputs from the interface unit.

Since the interface unit data bus had no signal on it, the buffers interpreted this as a high (or a 1). This high caused errors because at times it was transferred to the Z-80, overriding the low (or 0) that should have been there. Therefore, in the final design presented here the data bus is buffered in one direction only—out to the interface unit. Unfortunately, this results in two data buses in the interface unit: one in (not buffered) and one out (buffered).

This type of arrangement is not uncommon, of course, and presents no problems, except for some additional wiring. Buffered lines are denoted by the "B" following the line designation (e.g., D6B means that data line 6 has been buffered).

The interface board is connected to the TRS-80 by means of a 40-wire cable. At the TRS-80 end an AMP P/N 88103-I card edge connector (or equivalent) is required. At the interface board end I chose to cable directly to the sockets holding the ICs.

This has presented no problems, but connection could be made to the interface board card connector if desired (if enough pins are available). While two 20-wire ribbon cables would seem desirable and easier to connect on the AMP P/N 88103-I, my unit works successfully using regular 40-wire cable about 12 inches long.

The Interface Board itself can be described in four separate sections:

Data Bus Buffers. The data bus (D0-D7) is buffered by ICs a and b. As mentioned earlier, only data to be output is buffered; input data is presented directly to the TRS-80 without buffering. The buffers are enabled by IC g, which provides a low signal whenever the $\overline{OUTB}$ or $\overline{WRB}$ control lines go low. To avoid overloading these buffers, no more than 40 output ports should be used unless additional buffering is provided.

Address Bus Buffers. The address lines (A0-A15) are buffered in much the same way es the data lines. The buffers are enabled by IC h on receipt of a low signal from any one of the four control lines $\overline{OUTB}$, $\overline{WRB}$, $\overline{INB}$, $\overline{RDB}$.

Control Lines. The remaining lines from the TRS-80 are whet I refer to as control lines. Once again I chose to buffer the control lines that are output from

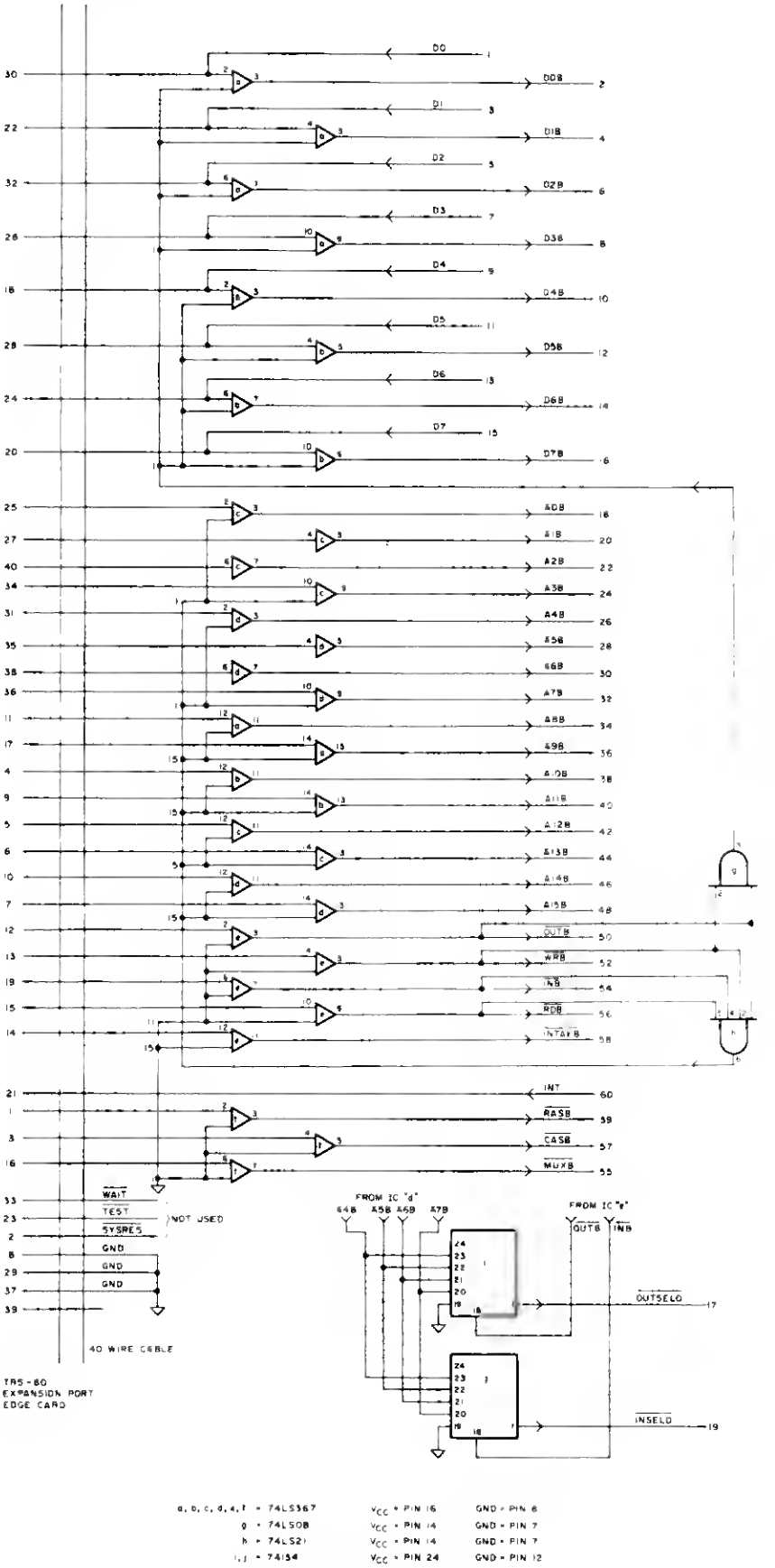


Fig. 1. Interface board.

the CPU; the one input line, INT, is not buffered. I decided not to buffer the $\overline{WAIT}$, $\overline{TEST}$ and $\overline{SYSRES}$ lines, since I could foresee no use for them in the near or even distant future; however, I wired them to the interface board just in case, so they could be buffered if desired in the same way as other lines.

The control lines are buffered by ICs e and f. Note that there are three GND lines. These should be connected to the GND of the interface board power supply. The line connected to pin 39 of the TRS-80 edge connector warrants mention here.

In the Level I manual (page 228) this line is shown connected to +5 V in the TRS-80. Prior to having my Level II installed, this was, in fact, the case; however, after the installation of the Level II, I noticed that the land to pin 39 had been cut and pin 40 hed been strapped to pin 39, making pin 39 a ground line.

Since I do not know the state of other units with regard to this pin, I recommend that this pin not be wired. Of the nine control lines wired, only two are used by circuits described in this article —the OUT and IN lines—however, the board has been designed to allow for the easy addition of memory and an interrupt board at a later date, which require the additional control lines.

Output and Input Port Initial Selectors. Whenever the TRS-80 executes an input or output (port) instruction, the port address is placed on the lower eight bits of the address bus (A0-A7). At the same time, the $\overline{OUTB}$ line (on an OUT instruction) or the $\overline{INB}$ line (on an INP instruction) is enabled. The input or output port initial selectors (IC i or j) are selected by these lines. This causes the high four bits (A4-A7) to be decoded by the selected IC i or j, which are 74154, four line to 16-line decoders.

The output of the 74154 is used to select a particular input or output board where the final port address is decoded. Thus using this configuration, up to 16 input and 16 output boards could be selected, providing additional buffers are used.

In the design presented here only one input and one output board is used, each one containing ports 0-15. To select additional boards, simply use the proper output from the 74154s to select the desired board (e.g., to select ports 16-31, the output from pin 2 of the relative 74154 would be used).

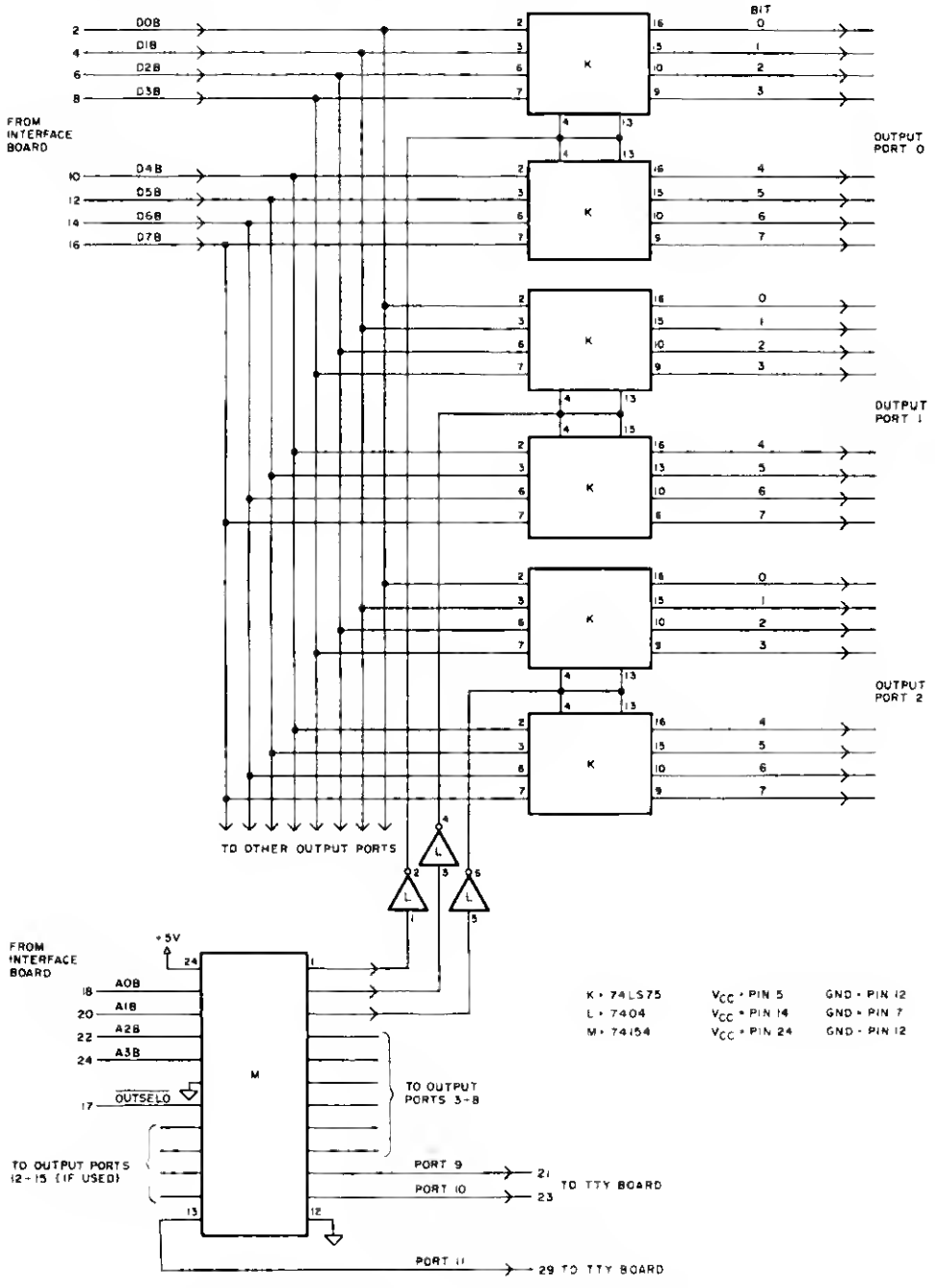


Fig.2. Eight-bit parallel output port board (only three ports shown).

Output Port Board

The output port board (see Fig. 2) provides up to 16 8-bit parallel output ports. In my configuration I have used ports 0 to 8, since this was the physical limit of the size of the board I have available.

The board is selected by the $\overline{OUTSEL0}$ line from the interface board. This enables IC ma 74154, which now decodes the four bits presented to it on the A0B-A3B lines.

The output from IC m is a low on one of the 16 output lines, corresponding to the binary value of lines A0B-A3B. This low is inverted and subsequently enables a pair of 74LS75 quad latches. The data on the D0B-D7B bus is now latched by the 74LS75 quad latches. The true data is now held by the latch and can be used to control external devices. The use of the edge card connector pins is left to the discretion of the user.

Input Port Board

The input port board (Fig 3) operates in a similar fashion to the output port board. The input board is selected by the $\overline{INSEL0}$ line from the interface board, enabling the 74154 to decode the final port address, according to the data on the A0B-A3B lines. The output from the 74154

(IC o) strobes the selected input port (IC n), and the data present on the input lines is transferred to the data bus. Again, due to physical limitations, my board only has nine input ports.

Either 74LS367 or 74LS368s can be used as the input port; the pinout for either is identical. The only difference is that the 74LS368 inverts the data present on the input lines, whereas the 74LS367 does not. This can be useful.

Imagine a port (x) with 74LS367s and only one input, bit 0, on that port being used. Performing a y = Inp(x) instruction will result in y having a value of 254 or 255, depending on whether the input is high or low. The other seven inputs are seen as high by the TRS-80.

However, if 74LS368s are used, then the highs on bits 1-7 will be inverted and seen by the TRS-80 as low or 0. Consequently, y will now have a value of 0 or 1. This does make programming a little easier.

Now that we can input and output to the TRS-80, a whole new world has opened up! The following are three of the uses that I have successfully tried to date, obviously there are many more.

TTY Interface

Shortly after completing the interface board, I purchased a Model 33 at almost bargain-basement price (even in our devalued Canadian dollars!) from a local dealer at a clearout sale.

I constructed the TTY board (see Fig. 4) in a couple of evenings. It uses a popular UART, the AY-3-1015, and was selected primarily because of the single 5 V supply required. Other UARTs would probably work just as well. Whichever UART you purchase, I suggest you obtain a copy of the specification sheets, as many variations are allowed (e.g., parity, number of stop bits, number of bits/character, etc).

To list the numerous variations here would be too lengthy; however, the circuit as shown will run a Model 33 Teletype at 110 baud, 20 mA current loop in half duplex operation. No programming is necessary to convert the serial data to parallel or vice versa, as this is done by the UART.

The 555 timer circuit supplies clock pulses at 16X the desired baud rate; therefore, the clock frequency for 110 baud is 1760 Hz. The actual serial data is transmitted to and received from the TTY by the two 4N26 optical couplers and the 2N2222 transistors. These couplers provide electrical isolation between the TTY and UART.

At the start of any program that will input or output data through the UART, an OUT 11,0 instruction should be used to reset all internal UART registers and flags to 0.

To input data from the TTY, an INP9, (x) instruction will enable the $\overline{\text{SWE}}$-0 line (status word enable). If bit 1 is a 1, the DAV (data available flag) line will be high, indicating that the UART does, in fact, have data to input. An INPIO(x) will enable the $\overline{\text{RDE}}$ line (received data enable) and will result in the data being placed onto the data bus.

Following this, an OUT 10,0 should be executed to enable the $\overline{\text{RDAV}}$ line (reset data available flag). Obviously, to ensure that no input is missed, these instructions should be contained in a loop, with a branch out only when a character is read.

To output data to the TTY an INP9,(x) instruction will again result in the status word being output on the data bus. This time, however, we are interested in bit 0, which will be the TBMT flag (transmitter buffer empty). If the TBMT flag is a 1 then the data may be output to the UART for transmission. To do this, an OUT9,(x) instruction is required.

Note that during transmission from the UART the EOC line on pin 24 goes low. This keeps

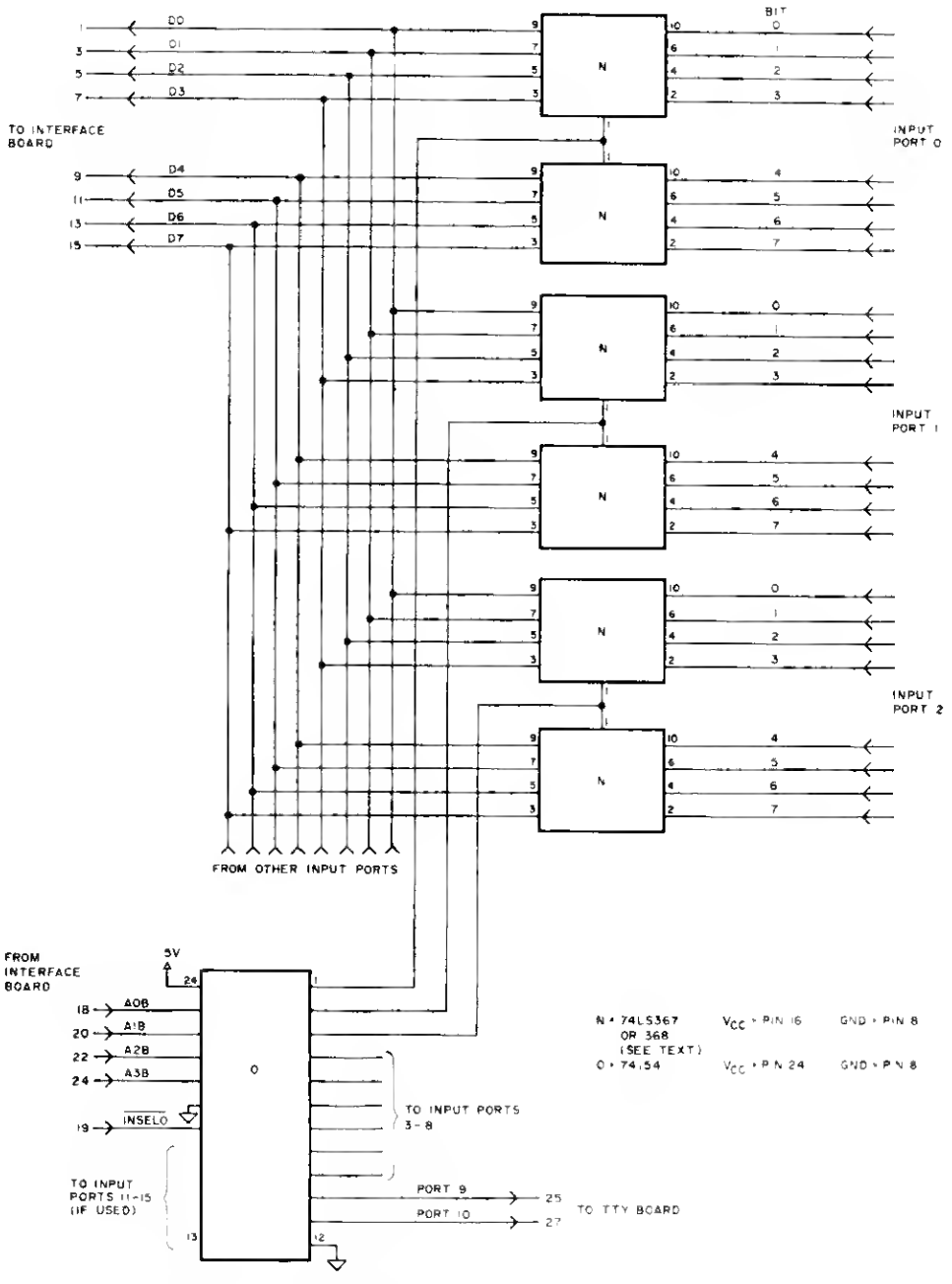


Fig. 3. Eight-bit parallel input port board (only three ports shown).

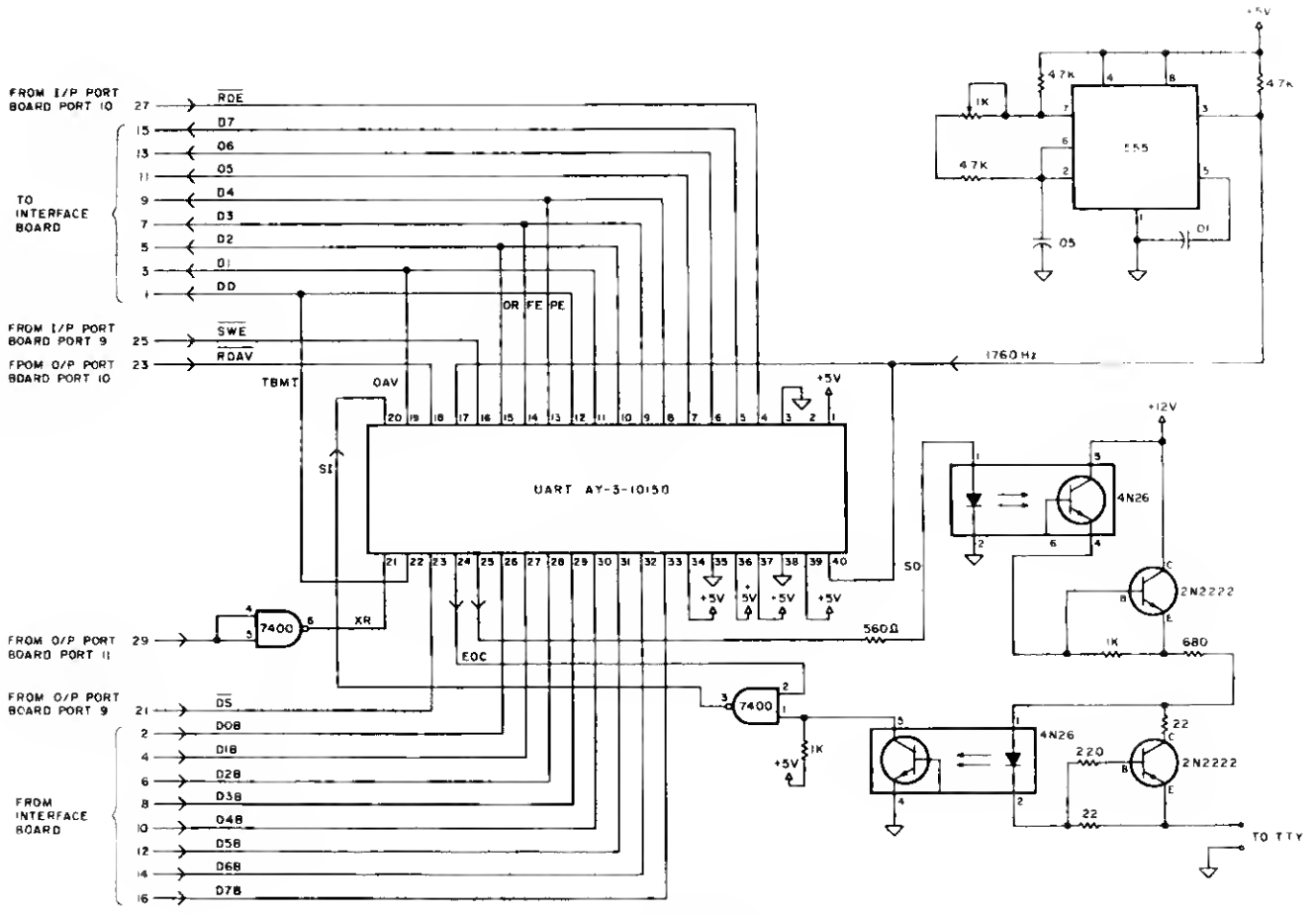


Fig. 4. TTY interface.

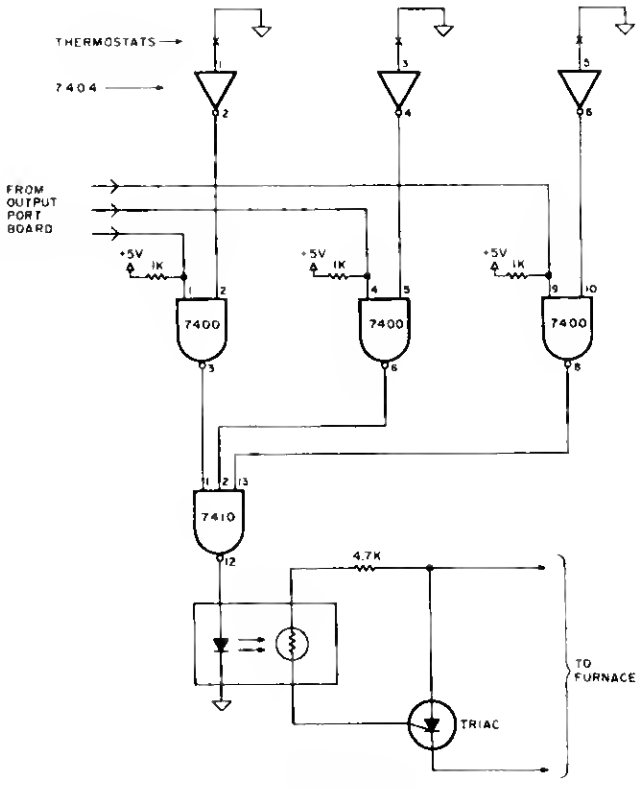


Fig. 5. Furnace control.

the output from the 7400 high, which prevents the UART from seeing the transmitted character on the receiver side (pin 20).

Three other flags are output on the data bus: the OR (overrun), FE (framing error) and PE (parity error). These can be checked by software if required, but this is not absolutely necessary.

Home Climate Controls

Programmable timers that will turn down the thermostat setting at night are available, however, the cost of two more thermostats is even less, and besides that, it gives your TRS-80 something to do while you are working!

My house has three levels, and so a thermostat on each level is enabled by a signal from an output port under program control. As a safety feature, I have wired three outputs through a plug and socket arrangement, so that in the event of a failure of the computer, by disconnecting the plug, all three thermostats are automatically enabled (see Fig. 5). (There's nothing worse than trying to fix a program bug when you can't see the monitor for the ice crystals!)

The triac—I used one from my junk box, as the voltage and current demands are minimal (check this on your unit)—turns on the furnace as the thermostat used to do. The triac is turned on by a simple circuit consisting of a LED and an LDR (light dependant resistor), which I bought at Radio Shack. Of course, these two items must be enclosed in a lightproof container to be effective.

To provide a means of keeping time in the computer, I used a one-minute pulse from a digital clock (which I had built some time ago) connected to input port 0. The clock itself is driven by a 160 kHz crystal, which is divided by a number of binary counters (7493s) connected in series to produce a one-minute pulse. The input port

```
                00005 ;****************** TTYOP1 ******************
                00010 ;TTYOP1 ALLOWS USE OF A REGULAR TTY WITH THE
                00020 ;TRS-80. THIS ALLOWS DIRECT USE OF LLIST AND
                00030 ;LPRINT COMMANDS. IT RESIDES AT 7F00(H) WHICH
                00040 ;IS THE ADDRESS (32512D) THAT MUST BE ANSWERED
                00050 ;IN RESPONSE TO "MEMORY SIZE?". AFTER LOADING
                00060 ;A "/" WILL LOAD THE DCB (4026H & 4027H) WITH
                00070 ;THE "START" ADDRESS AND WILL THEN JUMP TO
                00080 ;BASIC
                00090
7F00            00100          ORG     7F00H
7F00 D308       00110          OUT     (11),A          ;RESET UART
7F02 21107F     00120          LD      HL,START        ;ADDR OF TTYOP1
7F05 222640     00130          LD      (4026H),HL      ;INTO DCB
7F08 212A40     00140          LD      HL,402AH        ;CHAR COUNT ADDR
7F0B 3640       00150          LD      (HL),64         ;LOAD # OF CHAR/LINE
7F0D C3191A     00160          JP      1A19H           ;JP TO BASIC
7F10 79         00170 START    LD      A,C             ;CHAR TO BE O/P
7F11 FE0D       00180          CP      13              ;CK IF CR
7F13 2004       00190          JR      NZ,A1           ;JP IF NOT
7F15 CD2F7F     00200          CALL    CRLF            ;CR+LF ROUTINE
7F18 C9         00210          RET
7F19 CD267F     00220 A1       CALL    OPCHAR          ;CHAR O/P ROUTINE
7F1C DD7E05     00230          LD      A,(IX+5)        ;LD # OF CHAR LEFT
7F1F FE00       00240          CP      0               ;CK IF CRLF NEEDED
7F21 C0         00250          RET     NZ              ;RET IF NOT
7F22 CD2F7F     00260          CALL    CRLF            ;CR+LF ROUTINE
7F25 C9         00270          RET
7F26 CD3E7F     00280 OPCHAR   CALL    CKT8MT          ;CK IF T8MT
7F29 D309       00290          OUT     (09),A          ;O/P CHAR
7F2B DD3505     00300          DEC     (IX+5)          ;DEC CHAR COUNTER
7F2E C9         00310          RET
7F2F 3E0D       00320 CRLF     LD      A,13            ;LOAD CR
7F31 CD267F     00330          CALL    OPCHAR          ;O/P CHAR
7F34 3E0A       00340          LD      A,10            ;LOAD LF
7F36 CD267F     00350          CALL OPCHAR             ;O/P LF
7F39 DD360540   00360          LD      (IX+5),64       ;RELOAD CHAR COUNTER
7F3D C9         00370          RET
7F3E F5         00380 CKT8MT   PUSH    AF              ;SAVE CHAR IN A
7F3F DB09       00390          IN      A,(09)          ;I/P UART FLAGS
7F41 E601       00400          AND     1               ;STRIP OFF T8MT
7F43 FE01       00410          CP      1               ;CK IF MT
7F45 20F8       00420          JR      NZ,CKT8MT+1     ;JP IF NOT
7F47 F1         00430          POP     AF              ;RESTORE A REG
7F48 C9         00440          RET
7F00            00450          END     7F00H
00000 TOTAL ERRORS

CKT8MT  7F3E
OPCHAR  7F26
CRLF    7F2F
A1      7F19
START   7F10
```

Program A.

is continously monitored for a change in state. Other methods could be used, e.g., a FOR-NEXT loop or a 555 timer circuit if you are not too concerned about accuracy.

Model Railway Speed Control

The speed control shown in Fig. 6 is a simple digital to analog converter circuit. With bit 3 low, the output of the converter circuit is low, hence Q1 and Q2 are turned off. With bit 3 high, a voltage is presented to the base of Q1, turning it and Q2 on. The exact voltage is determined by the binary value of bits 0, 1 and 2. The output voltage appearing at the emitter of Q2 is incremented in eight steps by decrementing the binary value of the four inputs to the 7406 (bits 0-3).

Perhaps the easiest way to explain this is by saying that with a value of 8, Q1 and Q2 are off and with a value of 0, they are full on. Thus, the train is stopped with a value of 8 and runs at its fastest speed with a value of 0 presented to the converter circuit from the output port. For values between 0 and 7, the train runs at a correspondingly slower speed. The actual voltage is from about 6 V, which is the lowest voltage that most HO-scale trains will run at, to about 11.5 V (assuming a 12 V supply).

Software

If you decide to build the TTY interface board, the following programs should greatly enhance the capabilities of your computer.

Programs A and B allow the use of the resident TRS-80 LLIST and LPRINT commands with a model 33 (or similar) TTY and the TTY interface board previously described.

The TRS-80 is designed to produce hard copy on a line printer through a memory mapped I/O port at address 14312 (37E8H). The software routines necessary to permit this function are continued within the BASIC ROM.

I first thought that I would be able to use these routines by decoding address 14312 and wiring the UART circuit to it. However, I found that the ROM routines do not issue a line feed command, at the end of a line of print only a carriage return command is issued. Obviously the Radio Shack line printer automatically line feeds whenever it receives a carriage return. A 33 TTY does not!

With the help of the RSM monitor I eventually found the answer.

On power-up initialization a number of addresses in RAM are loaded with information used by the BASIC interpreter. Two of these addresses 16422 and 16423 (4026H & 4027H) are loaded with the entry point of the line printer output routine—14212 (0580H). By providing my own TTY handling routine and directing the BASIC interpreter to it by changing the contents of 16422 and 16423, the TRS-80 can output to TTY rather than to the line printer.

Programs A and B do just that. Program A is the actual assembly language program which I produced using the Radio Shack Editor Assembler. It generates a line feed whenever a carriage return is performed. It also generates a CR and LF when 64 characters are printed on any line, thus the hard copy looks exactly the same as displayed on the monitor.

If you have the Editor Assembler program, I recommend producing program A and making a tape copy of it. Simply load it using the system command and enter a "/". This loads the pointer addresses 16422 & 16423 and returns to BASIC.

For those who do not have the Editor Assembler, program B is provided. This is a BASIC language program which POKEs the machine (or assembly) language program into high memory. Once POKEd, the BASIC program can be deleted and the TTY handler program will remain in high memory until power is removed.

Two versions are shown: one for 4K and one for 16K. Remember that whatever method you use, the memory size must be

```
10 REM THIS IS THE 4K VERSION OF A BASIC PROGRAM FOR LOADING
20 REM A MACHINE LANGUAGE PROGRAM INTO HIGH MEMORY TO ALLOW
30 REM USE OF A TTY WITH LLIST AND LPRINT COMMANDS.
40 REM ONCE LOADED, THE TTY HANDLER WILL REMAIN IN MEMORY
50 REM UNTIL POWER IS REMOVED, AND THIS PROGRAM MAY BE ERASED
60 REM MEMORY SIZE MUST BE SET AT 20224 PRIOR TO RUNNING THIS
70 REM PROGRAM. OUTPUT TO THE TTY IS THROUGH PORT 9
90 CLS
100 FORX=20224TO20296
110 READY:POKEX,Y
120 NEXT
130 POKE16526,0:POKE16527,79
135 PRINT"TTY HANDLER LOADED"
140 X=USR(0)
150 END
1000 DATA211,11,33,16,79,34,38,64,33,42,64,54,64,195,25,26,121
1010 DATA254,13,32,4,205,47,79,201,205,38,79,221,126,5,254,0
1020 DATA192,205,47,79,201,205,62,79,211,9,221,53,5,201,62,13
1030 DATA205,38,79,62,10,205,38,79,221,54,5,64,201,245,219,9
1040 DATA230,1,254,1,32,248,241,201
```

Program B.

Bit	Meaning
0	Always 0 (TBMT)
1 = 1	DAV (data available)
2 = 1	OV (overrun error)
3 = 1	FE (framing error)
4 = 1	PE (parity error)

Table 1.

set to 20224 for a 4K computer or 32512 for a 16K computer. If you use program B take care when entering the DATA statements. One wrong entry will probably cause your computer to get lost which will require a Power-Off reset to get it back, which will erase your program entirely.

Program C. Program C is a TTY test and demonstration program. It initially requests the operator to input the number of "fox" messages required and then goes on to output the number of times requested the standard TTY test message: "The quick brown fox jumps over the lazy dog. 0123456789".

The operator is then prompted to type a message. Note that the message is terminated with a semicolon (;). The typed letters are displayed on the monitor screen and are also typed back on the TTY providing that no error flags are set.

Thus, if you have a suspect TTY, the location of the problem can be determined (i.e., keyboard or printing unit) by using this program. For example; if the "fox" message types OK and the characters displayed on the monitor are incorrect, then obviously the trouble is in the keyboard or transmitter portion of the TTY. Of course, the UART wiring is also checked by this test.

Line 1090 is a line in a continuous loop monitoring the status word flags for a change. If a change in state on any flag except the TBMT flag is detected, line 1100 will determine whether an error is present with the received character. If an error exists, then a transfer will be made to line 1150, where the particular error is determined. If no error has been detected by the UART, control will drop through to line 1110, where the received character is processed.

To save typing and memory, lines 1150-1195 can be replaced by:

```
1150 A$="TTY ERROR, FAULT CODE":S$
   =STR$(S):A$=A$+S$
1160 PRINTA$:GOSUB1300
```

In this case, you must break down the decimal fault code given into binary and Table 1 used to determine the error.

```
1000 CLEAR500
1010 INPUT"# OF FOX MESSAGES";X
1020 OUT11,0:GOSUB1330
1023 A$="TTY TEST PROGRAM":GOSUB1300
1026 A$="THE QUICK BROWN FOX JUMPS OVER THE LAZY GOG.0123456789"
1030 FORZ=1TOX
1035 GOSUB1300
1040 NEXT
1050 A$="PLEASE TYPE A MESSAGE":GOSUB1300
1070 CLS
1080 A$=""
1090 S=INP(9):S=SAND30:IFS=0THEN1090
1100 IFS>2THEN1150
1110 A=INP(10):OUT10,0:IFA=59THEN1130:REM ; TERMINATES INPUT
1120 X$=CHR$(A):A$=A$+X$:PRINT@256,A$;:GOTO1090
1130 GOSUB1330
1131 GOSUB1300
1135 A$="TTY INPUT LOOKS OK-PLEASE TYPE AGAIN":GOSUB1300
1137 GOTO1070
1150 S5=S:S5=S5AND16:IFS5<>0THEN1180
1160 S5=S:S5=S5AND8:IFS5<>0THEN1190
1170 S5=S:S5=S5AND4:IFS5<>0THEN1195
1171 GOTO1197
1180 A$="PARITY ERROR":PRINTA$:GOSUB1300
1182 GOTO1160
1190 A$="FRAMING ERROR":PRINTA$:GOSUB1300
1192 GOTO1170
1195 A$="OVERRUN ERROR":PRINTA$:GOSUB1300
1197 A$="TYPE AGAIN":PRINTA$:GOSUB1300
1200 FORZ=1TO100:NEXT
1210 OUT11,0:GOTO1080
1300 FORX=1TOLEN(A$)
1310 C$=MID$(A$,X,1):C=ASC(C$):GOSUB1360
1320 NEXT
1330 C=13:GOSUB1360
1340 C=10:GOSUB1360
1350 RETURN
1360 S=INP(9):S=SAND1:IFS=0THEN1360
1370 OUT9,C:RETURN
```

Program C.

Conclusion

In addition to the uses already described, the interface unit has been used to turn on and off outside lighting at Christmastime, as a telephone dialer and is presently being used to control basement lighting in addition to the climate control system previously described.

Providing care and patience are used, even a novice should be able to build this unit, as no special tools are required. Once this unit is built, I am sure that you will discover that your TRS-80 is no longer just an "expensive toy," but rather a useful addition to your household.

Should you decide to build the unit and if you have any comments or suggestions, I would be pleased to hear them. ■

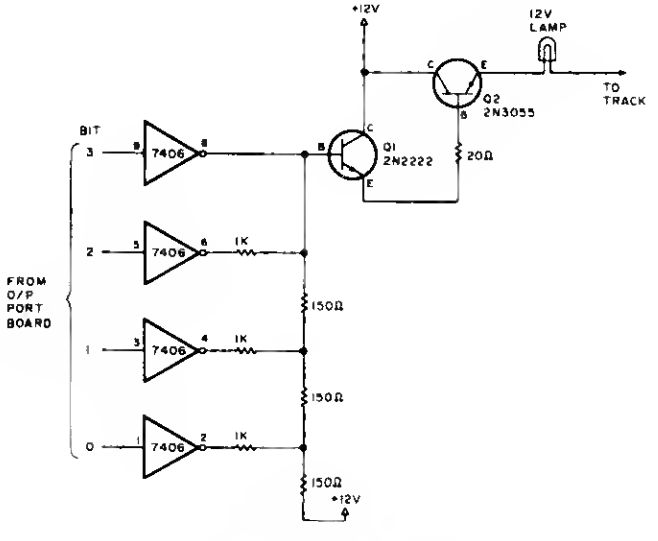


Fig. 6. Model railway speed control.

How to share your video information with your line printer.

LPRINT Routines

Craig Wemer
Abington Computing Group
1824 Watson Rd.
Abington PA 19001

```
10 INPUT"DO YOU HAVE A PRINTER(YES OR NO)";A$
20 IF A$="NO" THEN Q=1
100 C$=$$###### ##
110 PRINT"THE SALES TAX ON THE ITEM IS";USING C$;T
120 IF Q=1 THEN 130 ELSE LPRINT"THE SALES TAX ON THE ITEM IS";USING C$;T
130 PRINT"THE TOTAL COST IS";USING C$;C
140 IF Q=1 THEN 150 ELSE LPRINT"THE TOTAL COST IS";USING C$;C
150 PRINT"THE AMOUNT FINANCED IS";USING C$;F
160 IF Q=1 THEN 170 ELSE LPRINT"THE AMOUNT FINANCED IS";USING C$;F
170 PRINT"THE MONTHLY PAYMENT IS";USING C$;P
180 IF Q=1 THEN 190 ELSE LPRINT"THE MONTHLY PAYMENT IS";USING C$;P
190 REM THERE ARE 9 MORE STATEMENTS
```

Example 1. Repetition of statements.

Working with the TRS-80 microcomputer and its companion line printer presents many inherent problems because, unlike many terminals, the printer is not directly connected to the keyboard, so that data on the video screen will not be typed out onto the printer, and vice versa. There is also the problem of the TRS-80 executing an LPRINT statement when either the printer or the expansion interface, or both, are turned off, or when no printer exists.

In the first two cases, the computer will "lock up," but only until the respective hardware is turned on. The outcome of the last case is dependent on whether the computer is hooked up to an expansion interface. If it is not, then the lock-up can be broken by simply pressing the Reset button in back of the keyboard. If an interface is connected, then the Reset button will cause a power-up, and the resident program will be lost. (However, even though it is lost it is still retained in memory and can be accessed by using a recovery procedure not to be discussed here.)

Robbing Peter to Pay Paul

A direct link between the keyboard and the printer is possible, but only at the expense of the video display. It is accomplished by POKEing the contents of the printer driver addresses into the video driver addresses. These numbers, 141 in 16422 and 5 in 16423, when POKEd into 16414 and 16415, respectively (i.e., POKE 16414,141: POKE 16415,5), will cause all output to go to the line printer. *Nothing* will appear on the video screen. All functions, including the AUTO and EDIT functions, will continue to work as before; however, all keyboard input will be input blindly, since the input line will not be printed on the line printer until ENTER is pressed.

This technique can also be used backwards by POKEing the video commands into the printer addresses (POKE 16422,88 : POKE 16423,4). This will cause all LPRINTs to be printed on the video and ignored by the printer, becoming for all intents and purposes, PRINTs. It is useful in testing a program containing LPRINTs for text errors and formatting without us-

```
10 INPUT"DO YOU HAVE A PRINTER(YES OR NO)";A$
20 IF A$="NO" THEN Q=1
110 A$="THE SALES TAX ON THE ITEM IS";S=T GOSUB7000
130 A$="THE TOTAL COST IS" S=C:GOSUB7000
150 A$="THE AMOUNT FINANCED IS":S=F:GOSUB7000
170 A$="THE MONTHLY PAYMENT IS":S=P:GOSUB7000
190 REM THERE ARE 9 MORE STATEMENTS
7000 C$=$$######.##
7010 PRINTA$;USING C$;S
7020 IF Q=1 THEN RETURN
7030 LPRINT A$;USING C$;S
7040 RETURN
```

Example 2. Use of a printer subroutine.

ing paper or even the printer.

Duplication and Some Alternatives

In most cases, however, it is desired to have output to both the video and the line printer concurrently. The simplest way to do this is to type the line twice—once with PRINT, the other with LPRINT, as in Example 1. This is the most obvious approach, but it is a terrible waste of time and memory, especially with conditional LPRINTs and many statements, as in the example. With a four line subroutine, all duplication can be avoided, as in Example 2. In this particular instance, the modification saved 700 bytes.

Another alternative to duplication is direct conversion of all PRINTs to LPRINTs. The command code for PRINT is 178, while the LPRINT code is 175. To change all PRINTs to LPRINTs, use the command in Example 3. While this command is designed to be used in the direct mode, it is easily adaptable to the programming mode. To reverse the process and go from LPRINT to PRINT use the command in Example 4.

These two commands are complex for a reason. They will change only PRINT and LPRINT statements and will leave all programs pointers, line numbers, GOTOs, GOSUBs and variables untouched, and will automatically end at the end of the resident program if the program is less than 16K.

A More Viable Solution

These routines are useful, but in practice the most useful method of outputting to the line-printer is by PEEKing the video display memory addresses and outputting the information to the line printer. It is useful in that it retains the format of the screen and will support PRINT @s and near graphics, which are normally ignored.

For best results, the subroutine should be called after the screen is filled and all inputs are entered. It is also necessary to CLEAR 150 bytes of memory for string manipulation.

The Screen Printer subroutine is found in Example 5. It is ideal for a sample run of a program, and could be called every time the screen is filled. It would type out every character in its place, including graphics.

Lines 20000 and 20115 make it possible to control access to the subroutine. Because of those lines, if the line printer is either initially off or is turned off during execution, the program will RETURN and continue execution and will *not* lock-up. (The address 14312, the line printer address, will store a 255 when the line printer's power is off, a 223 when its power is on but its motor switch is off, a 191 when its I/O buffer is partially full and a 63 when it is ready to receive instructions.)

The routine need not be typed in with the program, nor need it remain an intrinsic part of the program, although it could. Rather, it can be written with high line numbers (20000 or 30000), stored on a separate tape and Appended to the end of the resident program.

To Append: Print the PEEK of 16633 for the LSB of the End-of-Program pointer and the PEEK of 16634 for the MSB, then subtract two from the LSB (unless the LSB is 0 or 1, in which case subtract one from the MSB and add 254 to the LSB). Take the new LSB and MSB and POKE 16548, LSB : POKE 16549, MSB. Then load the subroutine using CLOAD, and POKE 16548,233 (186 if a Disk is attached) : POKE 16549,66 (104 with a disk system), then type in CLEAR:RESTORE, and the program is ready to run.

After Appending, add all necessary GOSUBs. One additional tip is helpful—choose uncommon variables in the subroutine (AZ,ZQ,Q9, etc) so that they can be Appended indiscriminately without fear of conflicting with program variables.

A similar routine for outputting a certain N number of lines can also be used (see Example 6). As is, the routine will LPRINT only the entire previous line. To LPRINT more than one line, simply add 64 for each additional line to be LPRINTed to the second term of line 21000 (64 + POS(0)) and to line 21010 N2 = N1 + 63. For example, if three lines were to be LPRINTed, the above lines would read (192 + POS(0)) and N2 = N1 + 191, respectively.

To make it completely general, with the value of N to be INPUT, the above lines could be changed to (64*N + POS(0)) and N2 = N1 + 64*N – 1, respectively. It works because the memory locations 16416 and 16417 return the current cursor position, and the POS(0) term returns the cursor, as far as the program is concerned, to the beginning of the line, so it need not be reset.

This remains an incomplete list of printer applications; however, we have found that these five routines form a useful and, in fact, almost indispensable advantage when working with the line printer. These routines could be extended into a text editor in BASIC, using the INKEY function to type a text onto the screen and then using either a shifted key or the ENTER key (which returns an ASCII 10) to branch to the Screen printer subroutine and output the text to the printer. Further applications are limited only by the imagination of the programmer. (Note: I would like to thank Jeff Eisen, Gene Fred Wieland and Robin Salmansohn

```
20000 IF PEEK(14312)<>63 THEN RETURN
20010 LPRINT STRING$(64,"+")
20020 FOR N=15360TO16383 STEP 64
20030 Q=PEEK(N)
20040 IF Q>=127 AND Q<=191 THEN Q=42
20050 A$=CHR$(Q)
20060 FOR Q=1TO63
20070 Z=PEEK(N+Q)
20080 IF Z>=127 AND Z<=191 THEN Z=42
20090 B$=CHR$(Z)
20100 A$=A$+B$
20110 NEXT Q
20115 IF PEEK(14312)<>63THEN RETURN
20120 LPRINT" ":LPRINT A$
20130 NEXT N
20140 LPRINT STRING$(64,"+")
20150 FOR N=1TO3: LPRINT"  ":NEXT
20160 RETURN
```

Example 5. Screen Printer subroutine.

```
21000 N1=PEEK(16417)*256+PEEK(16416)+64+POS(0)+PEEK(16416)
21010 N2=N1+63
21020 A=PEEK(N1):IF A>=127 AND A<=191 THEN A=42
21030 A$=CHR$(A)
21040 N1=N1+1:FOR H=N1 TO N2
21050 B=PEEK(A):IF B>=127 AND B<=191 THEN B=42
21060 B$=CHR$(B)  A$=A$+B$
21070 NEXT H
21080 LPRINT A$
21090 RETURN
```

Example 6. Single line Screen Printer subroutine.

```
FOR N = 17128 TO 32768  IF PEEK(N) = 0 AND PEEK(N + 1) = 0 AND PEEK(N + 2) = 0
THEN END  ELSE IF PEEK(N) = 0 THEN N = N + 4  NEXT . ELSE IF PEEK(N) = 178 THEN
POKE N,175  NEXT : ELSE NEXT
```

Example 3.

```
FOR N = 17128 TO 32768  IF PEEK(N) = 0 AND PEEK(N + 1) = 0 AND PEEK(N + 2) = 0
THEN END  ELSE IF PEEK(N) = 0 THEN N = N + 4  NEXT  ELSE IF PEEK(N) = 175 THEN
POKE N,178  NEXT  ELSE NEXT
```

Example 4.

✓ Reader Service—see page 147

for their various contributions along the way.)

Addendum

Since I did my original research on this topic, a few developments have changed certain data found in this article. These changes fall into two categories.

1) When I researched the article TRSDOS 2.1 was the only Disk Operating System available. Since then Radio Shack has introduced TRSDOS 2.2 and TRSDOS 2.3 and they have been widely distributed. Because of memory allocation, the beginning of a BASIC program is not always found at 17129 as in Level II BASIC. This affects APPENDing instructions and the PRINT to LPRINT conversion routine.

The values in question are still stored in addresses 16633 and 16634 and should be PEEKed for each individual version. Values for TRSDOS 2.1 are found in this article, and the value for TRSDOS 2.2 and 2.3 are 36 and 106 respectively. It is a purely academic point, however, since TRSDOS 2.2 and 2.3 each have APPENDing instructions built in. Still, manual appending is handy to have when a full disk prevents the user from SAVEing a disk program in the TXT mode.

The change in the beginning of the program also affects the PRINT to LPRINT conversion. To use the BASIC conversion program contained in the article, it is necessary to change the beginning address of the FOR-NEXT loop to 26809 for use with TRSDOS 2.1 and 27171 for use with TRSDOS 2.3.

Another problem is the expansion of memory usually associated with a disk system. BASIC is either too slow or incapable of changing all addresses in such a system. The following machine language code is identical in function to Example 3 and accomplishes the same task about 50 times faster. It can be accessed via the system command in Level II BASIC, and the CMD"I", "filespec" mode in BASIC. IN 2.1 it should be loaded into high memory before entering BASIC and then accessed as a USR routine. My designated file name for LPRINT to PRINT conversion is LPTPCV/CMD and for PRINT to LPRINT is PTLPCV/CMD.

2) Not all printers, particularly the newer models, enter the same data in the driver address at 14312 when the Print Inhibit Switch is turned off. It will not affect the Screen Printer Subroutine in this article as 63 still designates ready to go and 255 still designates that the power is off, but depending upon the printer, certain signals in other states will change. ■

```
          00110          ;PTLPCV  PRINT TO LPRINT CONVERSION
          00111          ;TO CHANGE TO LPTPCV  LPRINT TO PRINT CONVERSION
          00112          ;CHANGE CHECK (LINE 290) TO 'CP 175'  AND
          00113          ;CHANGE SWITCH (LINE 330) TO 'LD (IX),178'
DD2A9A40  00120          LD IX,(40A4H)   ;BEGINNING OF BASIC
DD2B      00130          DEC IX
DD7E00    00140 START    LD A,(IX)
DD7E01    00150          LD A,(IX+1)
DD7E02    00160          LD A,(IX+2)
FE00      00170          CP 0            ;TEST FOR END OF PROGRAM
CA191A    00180          JP Z,1A19H      ;BACK TO BASIC
DD7E00    00190          LD A,(IX)
FE00      00200          CP 0            ;TEST FOR POINTER
200C      00210          JR NZ,CHECK
DDE5      00220          PUSH IX         ;EXCHANGE REGISTERS
E1        00230          POP HL
010500    00240          LD BC,5         ;SKIP POINTERS
09        00250          ADD HL,BC
E5        00260          PUSH HL
DDE1      00270          POP IX          ;SWAP BACK
18DF      00280          JR START
FEB2      00290 CHECK    CP 178          ;CHECK FOR PRINT
2004      00300          JR Z,SWITCH
DD23      00310          INC IX
18D7      00220          JR START
DD3600AF  00330 SWITCH   LD (IX),175     ;CHANGE TO LPRINT
DD23      00340          INC IX
18CF      00350          JR START
          00360          END
```

Example 7.

Software for the 80's

Enter the world of "80" programs from Instant Software. A world where you and your TRS-80 can rule duchies, play poker, learn speed reading, explore caves, design antennas, launch rockets, or simply pay your bills.

Final Frontier

SPACE TREK II Protect the quadrant from the invading Klingon warships. The Enterprise is equipped with phasers, photon torpedoes, impulse power, and warp drive. It's you alone and your TRS-80 Level I 4K, Level II 16K against the enemy. **Order No. 0002R $7.95.**

SPACE TREK III Let yourself go to the far ends of the solar system – and beyond. This package includes:
- **Stellar Wars** – Shoot down the Tie fighters and destroy the Death Star.
- **Planetary Lander** – Land your spacecraft and plant your flag across the solar system.

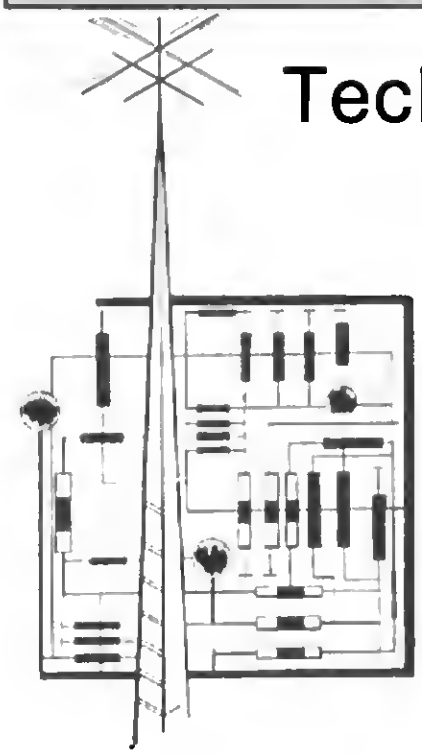
These one-player games require a TRS-80 Level I 4K. **Order No. 0031R $7.95.**

SPACE TREK IV Trade or wage war on a planetary scale. This package includes:
- **Stellar Wars** – Engage and destroy Tie fighters in your attack on the Death Star. For one player.
- **Population Simulation** – A two-player game where you control the economy of two neighboring planets.

You decide, guns or butter, with your TRS-80 Level II 16K. **Order No. 0034R $7.95.**

RAMROM PATROL/TIE FIGHTER/KLINGON CAPTURE Buck Rogers never had it so good. Engage in extraterrestrial warfare with:
- **Ramrom Patrol** – Destroy the Ramrom ships before they capture you.
- **Tie Fighter** – Destroy the enemy Tie fighters and become a hero of the rebellion.
- **Klingon Captura** – You must capture the Klingon ship intact. It's you and your TRS-80 Level II 16K battling across the galaxy. **Order No. 0028R $7.95.**

BASIC AND INTERMEDIATE LUNAR LANDER Bring your lander in under manual control. The basic version is for beginners, the intermediate version is more difficult, with a choice of landing areas and rugged terrain. For one player with a TRS-80 Level I 4K, Level II 16K. **Order No. 0001R $7.95.**



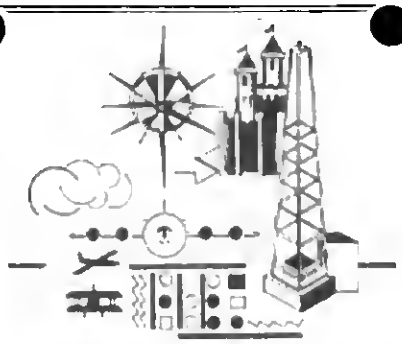
Technical Data

HAM PACKAGE I This versatile package lets you solve many of the problems commonly encountered in electronics design. With your Level I 4K or Level II 16K TRS-80, you have a choice of:
- **Basic Electronics with Voltage Divider** – Solve problems involving Ohm's Law, voltage dividers, and RC time constants.
- **Dipole and Yagi Antennas** – Design antennas easily, without tedious calculations.

This is the perfect package for any ham or technician. **Order No. 0007R $7.95.**

ELECTRONICS I This package will not only calculate the component values for you, but will also draw a schematic diagram. You'll need a TRS-80 Level I 4K, Level II 16K to use:
- **Tuned Circuits and Coil Winding** – Design tuned circuits without resorting to cumbersome tables and calculations.
- **555 Timer Circuits** – Quickly design astable or monostable timing circuits using this popular IC.
- **LM 381 Preamp Design** – Design IC preamps with this low-noise integrated circuit.

This package will reduce your designing time and let you build those circuits fast. **Order No. 0008R $7.95.**



Educational Games and Simulations

AIR FLIGHT SIMULATION Turn your TRS-80 into an airplane. You can practice takeoffs and landings with the benefit of full instrumentation. This one-player simulation requires a TRS-80 Level I 4K, Level II 16K. **Order No. 0017R $7.95.**

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. This program requires a TRS-80 Level I or II 16K. **Order No. 0043R $7.95.**

OIL TYCOON Avoid oil spills, blowouts and dry wells as you battle to become the world's richest oil tycoon. Two players become the owners of competing oil companies as they search for oil and control their companies. Requires a TRS-80 4K Level I or II. **Order No. 0023R $7.95.**

DOODLES AND DISPLAYS II Wait until your children get hold of this package:
- **Doodle Pad** – Drew pictures and save them on cassette tapes.
- **Symmetrics** – An electric kaleidoscope that changes from black to white and back again. It's almost hypnotic!
- **Drawing** – Like Doodle Pad, but for the serious artist. Over 40 user commands!
- **Random Pattern Display** – The computer does the drawing, but those with itchy fingers can tamper.
- **Mathcurves** – Bring those geometry lessons to life. Six different geometrical curves on the screen of your TRS-80.
- **Rugpattams** – Yes, it does design rug patterns; and with a choice of user or computer control, it can do a whole lot more.

For the Level II 16K TRS-80. **Order No. 0042R $7.95.**

WHO-DUN-IT? Criminal elements have committed five dastardly crimes. As the investigating detective, you must solve them.

You can compete against either Detective Nybbles, a computerized sleuth, or up to four other human detectives. Requires a TRS-80 Level II 16K.
- **Deduction** – Guess the order of four symbols out of six or seven different ones. To make things even more complicated, you can let the computer repeat symbols and have a range of 2401 possibilities. You'll need a TRS-80 Level II 4K. **Order No. 0047R $7.95.**

Reader Service—see page 147

GAMES

CARDS This one-player package will let you play cards with your TRS-80 – talk about a poker face!
•**Draw and Stud Poker** – These two programs will keep your game sharp.
•**No-Trump Bridge** – Play this popular game with your computer and develop your strategy.
This package's name says it all. Requires a TRS-80 Level II 16K. **Order No. 0063R $7.95.**

BEGINNER'S BACKGAMMON/KENO Why sit alone when you can play these fascinating games with your TRS-80?
•**Backgammon** – Play against the computer. Your TRS-80 will give you a steady, challenging game that's sure to sharpen your skills.
•**Keno** – Enjoy this popular Las Vegas gambling game. Guess the right numbers and win big.
You'll need a TRS-80 Level I or II. **Order No. 0004R $7.95.**

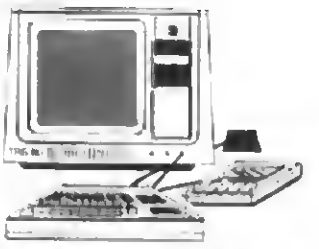
CAR RACE/RAT TRAP/ANTIAIRCRAFT Enjoy these challenging, fun-filled programs:
•**Car Race** – You and a friend can race on a choice of two tracks.
•**Rat Trap** – Trap the rat in his maze with your two cats. For one player.
•**Antiaircraft** – Aim and shoot down the enemy airplane. Requires Level I 4K TRS-80. **Order No. 0011R $7.95.**

DESTROY ALL SUBS/GUNBOATS/BOMBER This package of three programs is fun for the whole family. Included are:
•**Destroy All Subs** – Hunt down enemy subs while avoiding mines and torpedoes. A one-player game.
•**Gunboats** – Try to blow the enemy's ship out of the water. For one or two players.
•**Bomber** – Carefully release your bomb to destroy the moving submarine. A one-player game.
To enjoy these programs, you'll need a TRS-80 Level I 4K. **Order No. 0021R $7.95.**

BOWLING Let your TRS-80 set up the pins and keep score. One player can pick up spares and get strikes. For the TRS-80 Level I 4K, Level II 16K. **Order No. 0033R $7.95.**

GOLF/CROSS-OUT Have fun with these exciting one-player games. Included are:
•**Golf** – You won't need a mashie or putter – or a caddie, for that matter – to enjoy a challenging 16 holes.
•**Cross-Out** – Remove all but the center peg in this puzzle, and your neighbors will call you a genius.
You'll need a TRS-80 Level I 4K, Level II 16K. **Order No. 0009R $7.95.**

Utilities



TRS-80 UTILITY I Ever wonder how some programmers give their programs that professional look? Instant Software has the answer with the TRS-80 Utility I package. Included are:
•**RENUM** – Now you can easily renumber any Level II program to make room for modification or to clean up the listing.
•**DUPLIK** – This program will let you duplicate any BASIC, assembler, or machine-language program, verify the data, and record the program on tape. You can even do Level I programs on a Level II machine. For the TRS-80 Level II 16K. **Order No. 0081R $7.95.**

TRS-80 UTILITY II Let Instant Software change the drudgery of editing your programs into a quick, easy job. Included in this package are:
•**CFETCH** – Search through any Level II program tape and get the file names for all the programs. You can also merge BASIC programs with consecutive line numbers into one program.
•**CWRITE** – Combine subroutines that work in different memory locations into one program. This works with BASIC or machine-language programs and gives you a general checksum.
This package is just the thing for your TRS-80 Level II 16K. **Order No. 0076R $7.95.**

Education

VIDEO SPEED-READING TRAINER You can increase your reading speed and comprehension with this package. It uses the principle of the tachistoscope, a device that teaches by displaying images for a fraction of a second. This program can train you to recognize words and phrases quickly, so that your everyday reading becomes an uninterrupted process. To increase your throughput, you'll need a Level II 16K. **Order No. 0100R $7.95.**

MUSIC MASTER lets you compose music, play your keyboard as if it were a piano, and experiment with programming to produce music suited to your taste. This package includes:
•**Micro Organ** – The program will let you play flats and sharps to imitate the sounds of an organ, a harpsichord, or a piano.
•**Kaleidopy** – Now you can have a computerized "player piano." Generate a symmetrical graphics pattern and then see it transformed into music.
•**Composar** – Experiment with computer-generated music. This program allows you to select the length of the piece, the scale it will be played in, and the tempo.
•**Kaymanis** – This game will test not only your memory, but your musical ear. They may laugh when you sit down at the keyboard of your computer, but not after they hear what the Music Master package can do. All you will need is a TRS-80 Level II 4K. **Order No. 0084R $7.95.**

TYPING TEACHER This complete seven-part package takes you all the way from initial familiarization with the keys, through typing words and phrases, to complete mastery of the keyboard. Your computer can even become a bottomless page for typing practice. It requires a TRS-80 Level I 4K or Level II 16K. **Order No. 0099R $7.95.**

TEACHER

What do you need to learn? Would you like to know all of the cranial nerves? Electronic color codes? Civil War battles? Signs of the zodiac? Whatever your subject matter, the Teacher package can help you learn it. You simply input up to twenty questions and answers at one time. Next, review the material, and then take the test until you have your lesson down pat. The program gives you up to three hints per question and even offers graphic rewards for children, all at your discretion. All the information can be saved on cassette tape for reuse.

This package also contains the Teacher Data Transfer program, which allows you to combine several tests on one tape. That means you can learn a number of lessons sequentially without changing tapes.

Teacher is an effective instrument for anyone who needs to learn a lot of material in the shortest possible time. For decades leading educators and computer scientists alike have been promoting the future role of computers in education. Now you and your family can reap the benefits of computer-assisted instruction in your own home. The program is furnished with a blank data cassette tape. You'll need a TRS-80 Level II 16K. **Order No. 0065R $9.95.**

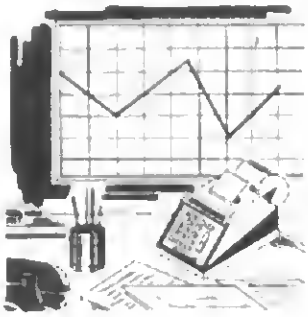
GRADE BOOK Teachers, now you can use the speed and accuracy of the TRS-80 to help you calculate student grades. Just type in the grades for tests, quizzes, homework, classwork, or special projects. The Grade Book program will calculate and display individual grade averages.

The program permits you to weigh student performance scores and convert raw score totals to a 100-points-equals-perfect-score basis.

You can also average students' quarterly grades with grades for the previous quarter, semester, and final exam to obtain an average grade for the year.

When grading time comes around, don't chain yourself to a calculator. Go modern with the Grade Book package. It requires a TRS-80 Level I 4K or Level II 16K. **Order No. 0050R $7.95.**

WORDWATCH This package combines four different programs to entertain and educate.
•**Word Race** – Here's a game for two would-be Grand Prix drivers who can define words accurately. The more you get right, the closer you come to the checkered flag.
•**Hide N Spell** – First you must find the misspelled word, then correct it. The faster you find it, the higher your score will be.
•**Spelling Bee** – This program is unique in that the student types back a spelling word in response to hearing it from your tape recorder. If the response is incorrect, hints and clues are given. Review up to 40 words in each session.
•**Spelling Tutor** – Load a spelling lesson, then sit back and observe as the computer does the rest. For variation the words are presented in different fashions, including reverse-order, with letters missing, and with altered letters.
There you have it: Wordplay x four = Wordwatch.
Requires a TRS-80 Level II 16K. **Order No. 0111R $7.95.**



Business

FINANCIAL ASSISTANT Compute the figures for a wide variety of business needs. Included are:
•**Depreciation** – This program lets you figure depreciation on equipment in five different ways.
•**Loan Amortization Schedule** – Merely enter a few essential factors, and your TRS-80 will display a complete breakdown of all costs and schedules of payment for any loan.
•**Financier** – This program performs thirteen common financial calculations. Easily handles calculations on investments, depreciation, and loans.
•**1% Forecasting** – Use this simple program to forecast sales, expenses, or any other historical data series.
All you need is a TRS-80 Level II 16K. **Order No. 0072R $7.95.**

BOWLING LEAGUE STATISTICS SYSTEM This package is the answer to the prayers of harried bowling league scorekeepers. The Bowling League Statistics System will keep a computerized list of league data, team data, and data for each bowler. It is extremely flexible and has a total of 16 different options to let you modify the program to suit your league's rules. The program is very easy to use and has extensive "built-in" aids to help you along. Requires TRS-80 Level II 16K. **Order No. 0056R $24.95.**

BUSINESS PACKAGE I Keep the books for a small business with your TRS-80 Level I 4K. The six programs included are:
•**General Information** – The instructions for using the package.
•**Fixed Asset Control** – This will give you a list of your fixed assets and term depreciation.
•**Detail Input** – This program lets you create and record your general ledger on tape for fast access.
•**Month and Year to Date Merge** – This program will take your monthly ledger data and give you a year to date ledger.
•**Profit and Loss** – With this program you can quickly get trial balance and profit-and-loss statements.
•**Year-End Balance** – This program will combine all your data from the profit-and-loss statements into a year-end balance sheet.
With this package, you can make your TRS-80 a working partner. **Order No. 0013R $29.95.**

BUSINESS PACKAGE III This package can change your TRS-80 into a full working partner for any businessman:
•**Inventory** – Maintain a computer-based inventory for a constant inventory system.
•**Commissions and Percentages** – Let your computer figure out markup and discount calculations, sales tax and more. This is a perfect time-saving package for any small business.
For the TRS-80 Level I 4K. **Order No. 0061R $7.95.**

BUSINESS PACKAGE IV This package can help any businessman get the right information for those critical decisions.
•**Business Cycle Analysis** – This program can plot the expansion and contraction cycles of any aspect of your business. You'll see in black and white just what's happening.
•**Financial Analysis** – Now you can get the figures for any type of annuity, sinking fund, or mortgage, and compute the yield and value for bonds.
The package includes a blank data tape and requires a TRS-80 Level I 4K or Level II 16K. **Order No. 0019R $9.95.**

*A trademark of Tandy Corporation

Home and Personal

HOUSEHOLD ACCOUNTANT Let your TRS-80 help you out with many of your daily household calculations. Save time and money with these fine programs:
•**Budget and Expense Analysis** – You can change budgeting into a more pleasant job with this program. With nine sections for income and expenses and the option for one- and three-month review or year totals, you can see where your money is going.
•**Life Insurance Cost Comparison** – Compare the costs of various life insurance policies. Find out the difference in price between term and whole life. This program can store and display up to six different results.
All you need is TRS-80 Level II 16K. **Order No. 0069R $7.95.**

PERSONAL FINANCE I Let your TRS-80 handle all the tedious details the next time you figure your finances:
•**Personal Finance I** – With this program you can control your incoming and outgoing expenses.
•**Checkbook** – Your TRS-80 can balance your checkbook and keep a detailed list of expenses for tax time.
This handy financial control for the home requires only a TRS-80 Level I 4K. **Order No. 0027R $7.95.**

STATUS OF HOMES/AUTO EXPENSES Two long-awaited programs that have got to save you money at work or in the home:
•**Status of Homes** – This program will allow you to keep track of all the expenses involved in building one house or an entire subdivision.
•**Auto Expenses** – Find out *exactly* what it costs you to drive your car or truck.
These programs require a TRS-80 Level I 4K. **Order No. 0012R $7.95.**

SEE INSIDE BACK COVER FOR LIST OF DEALERS NEAR YOU.

Order Your Instant Software today!

If the store nearest you does not stock Instant Software, use this order blank to purchase your software directly, or call Toll-Free 1-800-258-5473.

Name _____

Address _____

City _____ State _____ Zip _____

☐ Check ☐ Money Order ☐ VISA ☐ ☐

Credit card# _____ Exp date _____

Signature _____ date _____

Quantity	Order No.	Program name	Unit Cost	Total cost
		Shipping		$1.00
			Total order	

Mail to:
Instant Software™ ✓2
Peterborough NH 03458 USA Dept. 70C0

Prices Valid in USA Only

Smooth out your 80's power glitches with a simple regulator.

Regulate it!

William Klungle
1820 W. Lakewood Blvd.
Holland MI 49423

As steted many times before in *Microcomputing*, the Radio Shack TRS-80 is a lot of computer for the money invested. However, even with a good product such as the TRS-80, there is room for improvement.

One of the ereas that Radio Shack seems to have overlooked is the voltage reguletion of the monitor. The regulation in the computer itself is excellent, but voltage reguletion in the monitor is almost nonexistent. Any variation in the ac house current, such as may be caused by a pump or a dishwasher or a disk drive, results in a noticeeble fluctuetion of the video display.

Shortly after purchasing a TRS-80, I decided, for aesthetic reasons, to place the separate power module of the computer inside of the monitor case. This allowed the computer to reside on the family-room bookshelves and, with a small amount of rewiring, provided a single power switch for the entire system (see "Turn It Off!" *Microcomputing*, April '78, p. 114). As long as the monitor was on the workbench anyway, I took a close look at the power supply circuit to see what could be done about the regulation problem.

Regulating Transistor Circuit

The original circuit consisted of e half-wave rectifier and several RC filter networks (Fig. 1). The characteristics of the transistor circuits tend to amplify even the small variations in supply voltage, so that without some type of regulation the video display would never stand still.

In the monitor's early life as a portable television, there were provisions made on the chassis for an additional transistor to be mounted. The chassis has been punched to mount a TO-66-style transistor in the same area that the rectifier is mounted. Voltage regulation can easily be added by using only four inexpensive parts. The regulator circuit is not critical in its specifications, and any components that meet or exceed the minimum requirements may be used successfully. The original power supply provides approximately 120 V dc @ 350 mA. Any NPN silicon transistor in a TO-66-style case with a break-down voltage (VCEO) of over 150 volts and maximum current rating (Ic) of 500 mA should work.

Unfortunately, Radio Shack does not list some of the parts needed for this modification, so unless your local store happens to carry parts that are not in the catalog, you will have to seek another parts supplier. The parts I used are shown in Table 1.

The regulator circuit is wired as shown in Figs. 2 and 3. The 180k resistor serves as a current limiting resistor for the zener diode. The zener holds the base of the regulator transistor at 100 V dc. The transistor's emitter will always be within .6 volts

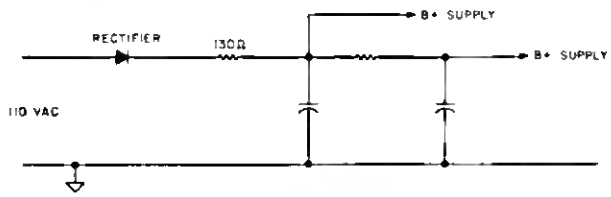


Fig. 1. Original circuit.

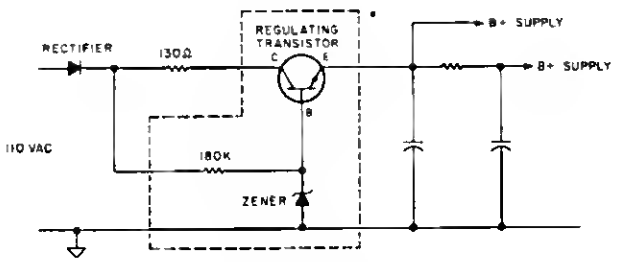


Fig. 2. Modified circuit.

1	Sylvania transistor	ECG 124
1	Sylvania socket	ECG 421
1	Sylvania zener diode	ECG 5050
1	180k 1W resistor	

Total cost should not exceed $5.

Table 1. Parts list

dc of the base voltage. The 130 Ohm, 7 Watt resistor, which was a part of the original power supply, distributes the supply voltage, which is in excess of the 100 volt output of the regulator.

Short the 22 Ohm resistor (jumper**) with a piece of wire. Removing this resistor allows the regulator to function over a greater range of line voltage variations.

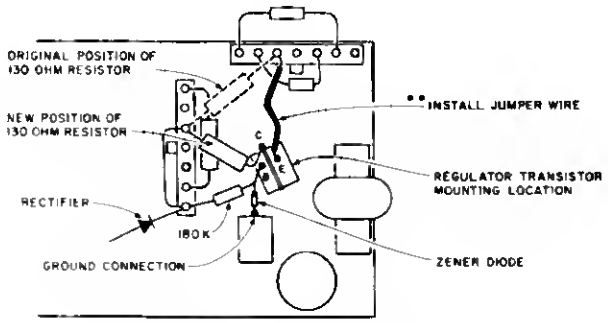


Fig. 3. Circuit modification.

Modification Tips

Consider the following possible hazards:

1. Be sure to unplug the power cord before you work on the monitor.

2. When installing the transistor, be sure to use the mica insulator and the two insulating washers supplied with the transistor. These isolate the transistor from the chassis.

3. Use a silicone-based heat-sink compound between the transistor and the mica and between the mica and the chassis. The silicone ensures proper heat dissipation.

4. Use caution when working around the exposed CRT (picture tube). A sharp blow on the neck of the tube could cause an implosion, which would be, at the least, costly—not to mention dangerous. Place a large towel or heavy cloth over the tube while it is exposed; this will protect you in case of accident.

5. Before putting the back on, turn the monitor on and check to make sure the raster is filling the screen. If not, adjust the centering rings located around the neck of the tube at the rear of the deflection yoke.

This entire project should take about one hour to complete and will put an end to the TRS-80's "dancing display."■

Forgotten which tapes are what? Use Whazit to identify system and BASIC tapes.

Whazit?

J. B. Penny
1537 Ramada
Houston, TX 77062

The Radio Shack TRS-80 computer with the Level II upgrade has two schemes for loading memory locations from cassette. The most frequently used is the CLOAD command. This reads in a tape in the BASIC format. A second scheme employs the SYSTEM command. This command is normally used to load machine language tapes.

Even though both methods operate at 500 baud, the formats are different. Any attempt to load using the wrong method will, at best, result in a simple failure to load. At worst, it could force you to reset or powerdown to regain control of the machine.

Deviations

To further complicate things, several available utility programs allow you to write cassettes in other than CLOAD or SYSTEM formats such as RSM-1S, GSF and The Electric Pencil. The Level II version of Microchess, for example, incorporates a special program loaded as a preamble to the main program.

Couple this with a typical hobby computerist's tape labeling and filing system and you'll get a fair sized mess. I must have a dozen tapes labeled simply "TEST". Some aren't labeled at all!

Machine code programs are particularly fond of deviations. Though the TRS-80 requires a file name when operating under the SYSTEM command, T-BUG allows you to load machine object code without one. But if you don't know where your "mystery" program is located, you may crash both programs.

Since I believe "it is better for me to light one small candle than to curse the darkness," I am furnishing the following listing. I call it WHAZIT. Though it's a long way from a total solution, WHAZIT can be a great help.

After loading WHAZIT I can read the header (and machine code trailer) from a 500 baud cassette without actually loading the tape into memory. The file name and memory location are displayed on the screen. Do yourself a favor; write it on the tape label this time.

What's WHAZIT?

WHAZIT was written on the Radio Shack Editor/Assembler. Its first 170 lines or so are the business end of the program.

After sync is found, the first eight bytes are read from the tape and stored in an assigned work space. A three way branch is then set up where the first byte after sync is checked to make a tentative format assignment. If neither Radio Shack format is found, the default message is printed on the screen. BTEST and STEST do further checking to confirm the format. If the format is BASIC, then the tape player is stopped and the file letter is printed on the screen along with the other information.

If the tape is in the SYSTEM format, then it is read all the way to the end to calculate the end address and extract the start point addresses. The start point address occupies the last two data bytes stored on the tape. This address is located when /ENTER is typed after loading a SYSTEM tape.

Clever programmers have been known to write TRS-80 program tapes where address blocks are not contiguous. The SYSTEM format allows this by preceding each data block with its own start address. If the addresses don't make sense, this may be the reason. Confusion may also be caused by a bad load.

Error checking is not included in the program.

Subroutines

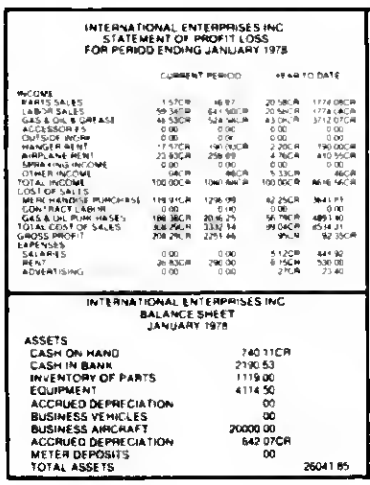
A couple of useful subroutines can be found buried in the program. COMPU will output whatever is loaded in the HL register as a four digit hexadecimal address. OUTPUT will print a string of ASCII characters, beginning with the address pointed to by the HL register plus one. The first byte should contain the length (number of characters) to be printed in the string. ■

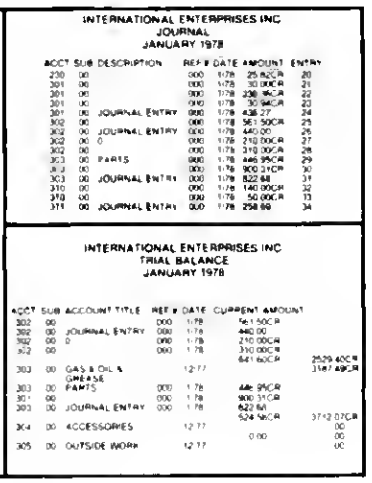
Program listing.

```
*A WHAZIT/HS
4CA8                00100          ORG     4CABH
                    00110 ;PROGRAM NAME WHAZIT: PRINTS FILE NAMES
                    00120 ;AND OTHER GOOD STUFF.  BY J. B. PENNY
4CAB 21003C         00130 START    LD      HL,3C00H; HOME CURSOR
4CAE 222D40         00140          LD      (40206),HL
4CB1 CD7C05         00150          CALL    5/CH        ; CLEAR SCREEN
4CB4 21FB4D         00160          LD      HL,MSG1
4CB7 CD3C4D         00170          CALL    OUTPUT
4CBA CD2B00         00180 KBD      CALL    28H         ;KEYBOARD SCAN
4CBD B7             00190          OR      A           ;FALLS THRU
4CBE 28FA           00200          JR      Z,KBD       ;IF CARRAGE
4CC0 FE0D           00210          CP      13          ;RETURN
4CC2 20F6           00220          JR      NZ,KBD
4CC4 AF             00230          XOR     A           ;MAKE A=0
4CC5 CD1202         00240          CALL    212H        ;DEFINE DRIVE
4CC8 CD9602         00250          CALL    296H        ;FIND SYNC
4CCB 21ED4D         00260          LD      HL,MBUFF
4CCE 0608           00270          LD      B,8
4CD0 CD3502         00280 LDLOOP   CALL    235H        ;READ 1ST 8
4CD3 77             00290          LD      (HL),A      ;BYTES INTO
4CD4 23             00300          INC     HL          ;MEM BUFFER
4CD5 10FF           00310          DJNZ    LDLOOP
4CD7 21ED4D         00320          LD      HL,MBUFF;BEGIN TEST
4CDA 3E55           00330          LD      A,55H       ;CHECK FOR
4CDC BE             00340          CP      (HL)        ;SYS. HEADER
4CDD 2835           00350          JR      Z,STEST
```

✓ Reader Service—see page 147

```
4CDF 3ED3     00360          LD      A,0D3H  ;CHECK FOR
4CE1 BE       00370          CP      (HL)    ;BASIC HEADR.
4CE2 2814     00380          JR      Z,BTEST
4CE4 21754E   00390 DEFALT   LD      HL,MSG2 ;DEFAULT MSG
4CE7 CD3C4D   00400          CALL    OUTPUT
4CEA ED5B2040 00410          LD      DE,(4020H);DESTINATION
4CEE 21ED4D   00420          LD      HL,HBOFF;  SOURCE
4CF1 010800   00430          LD      BC,8    ;  BYTE COUNT
4CF4 EDB0     00440          LDIR            ;  XFER DATA
4CF6 184D     00450          JR      TOFF
4CF8 23       00460 BTEST    INC     HL      ;TEST FOR 3
4CF9 BE       00470          CP      (HL)    ;D3'S IN A ROW
4CFA 20E8     00480          JR      NZ,DEFALT
4CFC 23       00490          INC     HL      ;DEFAULT IF
4CFD BE       00500          CP      (HL)    ;NOT FOUND
4CFE 20E4     00510          JR      NZ,DEFALT
4000 21164F   00520          LD      HL,MSG3
4003 CD3C4D   00530          CALL    OUTPUT
4006 21FD4D   00540          LD      HL,NBUFF+3
4009 7E       00550          LD      A,(HL)  ;PRINT THE
400A CD3300   00560          CALL    33H     ;FILE NAME
400D 3E22     00570          LD      A,22H   ;CLOSE
400F CD3300   00580          CALL    33H
4012 1831     00590          JR      TOFF
4014 CD3502   00600 BTEST    CALL    235H    ;READ BYTE
4017 47       00610          LD      B,A     ;COUNT & LD.
4018 CD864D   00620          CALL    HLADDR  ;START ADDR.
401B 22F54D   00630          LD      (STARTA),HL
401E 85       00640          ADD     A,L
401F 4F       00650          LD      C,A
4020 CD6F4D   00660          CALL    HLOCK
4023 CD3502   00670 LOOPD    CALL    235H
4026 FE78     00680          CP      78H     ;CHECK FOR ENTRY
4028 2855     00690          JR      Z,ENDSYS;  POINT HEADER
402A FE3C     00700          CP      3CH     ;CHECK FOR START
402C 20F5     00710          JR      NZ,LOOPD; OF DATA HEADER
402E CD3502   00720          CALL    235H
4031 47       00730          LD      B,A     ;BYTE COUNT
4032 CD664D   00740          CALL    HLADDR
4035 85       00750          ADD     A,L
4036 6F       00760          LD      C,A
4037 CD6F4D   00770          CALL    HLOCK
403A 1BE7     00780          JR      LOOPD
403C 46       00790 OUTPUT   LD      B,(HL)  ;PRINT A STRING
403D 23       00800          INC     HL      ;OF ASCII CHAR.
403E 7E       00810          LD      A,(HL)
403F CD3300   00820          CALL    33H
4042 10F9     00830          DJNZ    OUTPUT+1
4044 C9       00840          RET
4045 212020   00850 TOFF     LD      HL,2020H;  TURN OFF
4048 223E3C   00860          LD      (3C3EH),HL; BOTH D'S
404B CDF901   00870          CALL    1F9BH   ;  AND MOTOR
404E 216A4F   00880          LD      HL,MSG5
4051 CD3C4D   00890          CALL    OUTPUT
4054 CD2B00   00900 ASK      CALL    2BH     ;SCAN KEYBOARD
4057 B7       00910          OR      A       ;FOR BREAK OR
4058 28FA     00920          JR      Z,ASK   ;CARRAGE RET.
405A FE01     00930          CP      1       ;BREAK MEANS
405C CA6800   00940          JP      Z,668   ;RET. TO ROM
405F FE00     00950          CP      13      ;CARRAGE RET.
4061 CAAB4C   00960          JP      Z,START ;MEANS TO
4064 20EE     00970          JR      NZ,ASK  ;START OVER
4066 CD3502   00980 HLADDR   CALL    235H    ;READ NEXT 2
4069 6F       00990          LD      L,A     ;BYTES FROM
406A CD3502   01000          CALL    235H    ;TAPE & LOAD
406D 67       01010          LD      H,A     ;INTO THE HL
406E C9       01020          RET             ;REGISTER
406F B5       01030 HLOCK    ADD     A,L     ;READS A BLOCK
4070 6F       01040          LD      C,A     ;OF DATA UP TO
4071 CD3502   01050          CALL    235H    ;256 BYTES LONG
4074 23       01060          INC     HL      ;BUMP POINTER
4075 81       01070          ADD     A,C     ;KEEP CKSUM
4076 4F       01080          LD      C,A     ;IN REG. C
4077 CD2B00   01090          CALL    2BH     ;KBD SWEEP
407A FE01     01100          CP      1       ;STOP ON
407C 28C7     01110          JR      Z,TOFF  ;BREAK
407E 10F1     01120          DJNZ    HLOCK+2 ;LOOP TILL DONE
4080 CD3502   01130          CALL    235H
4083 B9       01140          CP      C       ;IS CKSUM VALID?
4084 CC2C02   01150          CALL    Z,22CH  ;THEN BLINK *
4087 C9       01160          RET
4088 2B       01170 ENDSYS   DEC     HL      ;STORE END ADDR.
4089 22F74D   01180          LD      (ENDA),HL
408C CD664D   01190          CALL    HLADDR  ;STORE ENTRY PT.
408F 22F94D   01200          LD      (ENTRYA),HL
4092 213C4F   01210          LD      HL,MSG4 ;SYSTEM MSG.
4095 CD3C4D   01220          CALL    OUTPUT
4098 0606     01230          LD      B,6     ;  PRINT
409A 21EE4D   01240          LD      HL,HBUFF+1; THE
409D 7E       01250 LOOPF    LD      A,(HL)  ;  FILE
409E CD3300   01260          CALL    33H     ;  NAME
40A1 23       01270          INC     HL
40A2 10F9     01280          DJNZ    LOOPF
40A4 21A84F   01290          LD      HL,MSG6
40A7 CD3C4D   01300          CALL    OUTPUT
40AA 2AF54D   01310          LD      HL,(STARTA)
40AD CDC44D   01320          CALL    COMPU   ;PRINT START ADDR.
40B0 CDCB4D   01330          CALL    COMMA
40B3 2AF74D   01340          LD      HL,(ENDA)
40B6 CDCA4D   01350          CALL    COMPU   ;PRINT END ADDR.
40B9 CDCB4D   01360          CALL    COMMA
40BC 2AF94D   01370          LD      HL,(ENTRYA)
40BF CDC44D   01380          CALL    COMPU   ;PRINT ENTRY ADDR.
40C2 1881     01390          JR      TOFF
40C4 7C       01400 COMPU    LD      A,H     ;WRITES HL REG.
40C5 CDD64D   01410          CALL    OUTHL   ;CONTENTS AS 4
40C8 7D       01420          LD      A,L     ;DIGIT HEX ADDR.
40C9 1808     01430          JR      OUTHL
40CB 3E2C     01440 COMMA    LD      A,2CH   ;PUTS IN COMMA
40CD CD3300   01450          CALL    33H
40D0 3E20     01460          LD      A,20H   ;AND SPACE
40D2 CD3300   01470          CALL    33H
40D5 C9       01480          RET
40D6 F5       01490 OUTHL    PUSH    AF
40D7 0F       01500          RRCA            ;SWAP PLACES
40D8 0F       01510          RRCA            ;WITH BITS 0-3
40D9 0F       01520          RRCA            ;AND BITS 4-7
40DA 0F       01530          RRCA
40DB CDDF4D   01540          CALL    BIASCI
40DE F1       01550          POP     AF      ;DO IT TWICE
40DF E60F     01560 BIASCI   AND     15      ;CLEAR BITS 4-7
40E1 FE0A     01570          CP      10      ;JUMP IF A<10
40E3 3802     01580          JR      C,NUMBER;IF A>9 THEN
40E5 C607     01590          ADD     A,7     ;CHANGE TO LTR.
40E7 C630     01600 NUMBER   ADD     A,'0'   ;ASCII OFFSET
40E9 CD3300   01610          CALL    33H
40EC C9       01620          RET
40ED 0000     01630 HBUFF    DEFW    0       ;FIRST 8 BYTES
40EF 0000     01640          DEFW    0       ;READ FROM TAPE
40F1 0000     01650          DEFW    0       ;ARE STORED HERE
40F3 0000     01660          DEFW    0
40F5 0000     01670 STARTA   DEFW    0       ;START ADDRESS HERE
40F7 0000     01680 ENDA     DEFW    0       ;END ADDRESS HERE
40F9 0000     01690 ENTRYA   DEFW    0       ;ENTRY ADDRESS HERE
              01695 *LIST OFF
00000 TOTAL ERRORS
READY CASSETTE
*P1695;*
01695 *LIST OFF
01700 MSG1     DEFB    121     ;SCREEN MSG8. HERE TO END
01710          DEFM    'THIS PROGRAM READS A 500 BAUD TAPE'
01720          DEFM    ' HEADER & PRINTS THE SYSTEM'
01730          DEFB    13
01740          DEFM    'FILE NAME OR BASIC FILE LETTER.  '
01750          DEFM    'LOAD TAPE AND HIT ENTER.'
01760          DEFM    0000H
01770 MSG2     DEFB    161
01780          DEFM    'THE HEADER JUST READ WILL NOT LOAD'
01790          DEFM    ' USING EITHER THE "SYSTEM"'
01800          DEFB    13
01810          DEFM    'COMMAND OR "CLOAD".  A GRAPHIC '
01820          DEFM    'REPRESENTATION OF THE FIRST'
01830          DEFB    13
01840          DEFM    'EIGHT BYTES FOLLOWING '
01850          DEFM    'THE SYNC CODE ARE '
01860 MSG3     DEFB    37
01870          DEFM    'HEADER IS IN BASIC.'
01880          DEFM    'FORMAT USE CLOAD '
01890 MSG4     DEFB    45
01900          DEFM    'HEADER IS IN "SYSTEM" FORMAT.  '
01910          DEFM    'FILE NAME IS  '
01920 MSG5     DEFB    61
01930          DEFB    13
01940          DEFM    'PRESS ENTER TO GO AGAIN, BREAK '
01950          DEFM    'TO QUIT AND RETURN TO BASIC.'
01960          DEFB    13
01970 MSG6     DEFB    62
01980          DEFB    13
01990          DEFM    'HEX STARTING, ENDING AND ENTRY PT. '
02000          DEFM    'ADDRESSES ARE AS FOLLOWS:'
02010          DEFB    13
02020          END     START
```



Find out how you are physically, mentally and emotionally.

Biorhythms

Ralph E. Holthausen
25 Willets Drive
Syosset, NY 11791

Even if you don't believe in biorhythms, your friends will enjoy this program when you draw their biorhythm curve for them. The program, created for the TRS-80 Level II BASIC, can be adapted to other BASIC units. For anyone who wants a numerical indication of the daily biorhythms there is a short alternate program included.

The idea behind the main program is thet biorhythms can best be represented by a sine curve. Normally, on peper, this curve is plotted on e horizontal axis. However, on the computer curve plotting is easier on a vertical axis. Highs or plus values are on the right of the vertical axis and lows or minus values are on the left.

The Theory

Biorhythm means rhythm of life. The basis of the biorhythm theory is that our lives are governed by cycles that start at our day of birth. There are three such cycles whose curves are plotted by the following program. The theory has been applied recently to accident prevention. Many companies are studying biorhythms and their effects on airline pilots, athletic teams; doctors and surgeons, relative to performing operations, have studied biorhythms.

From the day we are born, the theory states, we are governed by cycles. The 23 day cycle called the physical cycle governs the condition of one's body. The 28 day cycle called the emotional or sensitivity cycle governs one's temperament. The 33 day cycle, called the intellectual cycle, influences our intellect or thinking capacity. Those who have investigated the theory seem to agree on the length of these cycles as 23, 28 and 33 days respectively.

The physical cycle is said to affect our vitality and strength. The plus period which will be on the right side of the curve as printed by our computer, lasts 11½ days and these are days ot physical vitality, stamina, strength and durability. It is a period of self confidence, courage and progressive spirit. Athletes usually find this period

```
2   PRINT "              B I O R H Y T H M S                "
3   PRINT
4   PRINT "THERE ARE THREE CURVES...PHYSICAL, EMOTIONAL AND"
5   PRINT
6   PRINT "INTELLECTUAL, HIGHS ARE TO THE RIGHT....LOWS ARE"
7   PRINT
8   PRINT "TO THE LEFT.  FLASHER INDICATES YOUR BIORHYTHM FOR"
9   PRINT
10  PRINT "THIS DATE.  FLASHER ON LINE IS A CRITICAL."
11  PRINT
15  INPUT "ENTER YOUR BIRTHDATE..MONTH, DAY, YEAR"; M,D,YR
20  IF M <= 2 THEN 50
30  D1 = INT(365.25 * YR)
40  D2 = INT((M + 1) * 30.6) : GOTO 70
50  D1 = INT(365.25 * (YR - 1))
60  D2 = INT(M + 13) * 30.6
70  D3 = D + D1 + D2
80  INPUT "ENTER TODAY'S DATE..MONTH, DAY, YEAR" ; M,D,YR
90  IF M <= 2 THEN 120
100 D1 = INT(365.25 * YR)
110 D2 = INT((M + 1) * 30.6) : GOTO 140
120 D1 = INT(365.25 * (YR - 1))
130 D2 = INT((M + 13) * 30.6)
140 D4 = D + D1 + D2
150 DT = D4 - D3
160 P = INT(23 * (DT/23 - INT(DT/23)))
170 CLS
180 FOR I = Ø TO 2 * 3.14159265 STEP .48
190 PRINT TAB(20 * (1 + SIN(I))) ; "+"
200 NEXT I
210 X = 40
220 FOR Y = Ø TO 47
230 SET(X,Y)
240 NEXT Y
245 PRINT "PHYSICAL"
250 LET X1 = 40 * (1 + SIN( P * .273182))
260 LET Y1 = 2 + P * (39/23)
265 FOR N = Ø TO 200
270 SET(X1,Y1)
280 RESET(X1,Y1)
285 NEXT N
287 INPUT "DO YOU WISH TO SEE THE EMOTIONAL FOR TODAY" ;Z$
288 IF Z$ = "YES" THEN 300
290 END
300 E = INT(28 * (DT/28 - INT(DT/28)))
310 CLS
320 FOR I = Ø TO 2 * 3.14159265 STEP .48
330 PRINT TAB(20 * (1 + SIN(I))) ; "+"
340 NEXT I
350 X = 40
360 FOR Y = Ø TO 47
370 SET(X,Y)
380 NEXT Y
385 PRINT "EMOTIONAL"
390 LET X2 = 40 * (1 + SIN(E * .2244))
400 LET Y2 = 2 + E * (39/28)
405 FOR N = Ø TO 200
410 SET(X2,Y2)
420 RESET(X2,Y2)
425 NEXT N
427 INPUT "DO YOU WISH TO SEE THE INTELLECTUAL FOR TODAY" ;Z$
428 IF Z$ = "YES" THEN 500
430 END
500 L = INT(33 * (DT/33 - INT(DT/33)))
510 CLS
520 FOR I = Ø TO 2 * 3.14159265 STEP .48
530 PRINT TAB(20 * (1 + SIN(I))) ; "+"
540 NEXT I
550 X = 40
560 FOR Y = Ø TO 47
570 SET(X,Y)
580 NEXT Y
585 PRINT "INTELLECTUAL"
590 LET X3 = 40 * (1 + SIN(L * .1904))
600 LET Y3 = 2 + L * (39/33)
605 FOR N = Ø TO 200
610 SET(X3,Y3)
620 RESET(X3,Y3)
625 NEXT N
627 CLS
630 INPUT "DO YOU WISH ANOTHER BIORHYTHM" ;Z$
640 IF Z$ = "YES" GOTO 650
642 PRINT " I HOPE YOUR BIORHYTHMS WERE GOOD TODAY...GODD-BYE."
645 END
650 CLS : GOTO 2
```

Program Listing 1

Reader Service—see page 147

best for competitive sports.

The minus period (left side of the curve) also lasts 11½ days and is a period of reduced energy. One tires more easily, is more liable to infectious diseases; medicines seem to work well, according to authorities. This is a period of rejuvenation where our body seems to be recharging; a good period for rest and relaxation.

The emotional cycle affects our nervous systems. The plus period lasts 14 days and is a period of cheerfulness, creative ability and moral energy. This is a period where we are full of energy, good for contests, public performance, conducting jobs where teamwork is required.

The minus period also lasting 14 days is a period where we lack ambition, tend to be moody and should be careful in our personal relation with others.

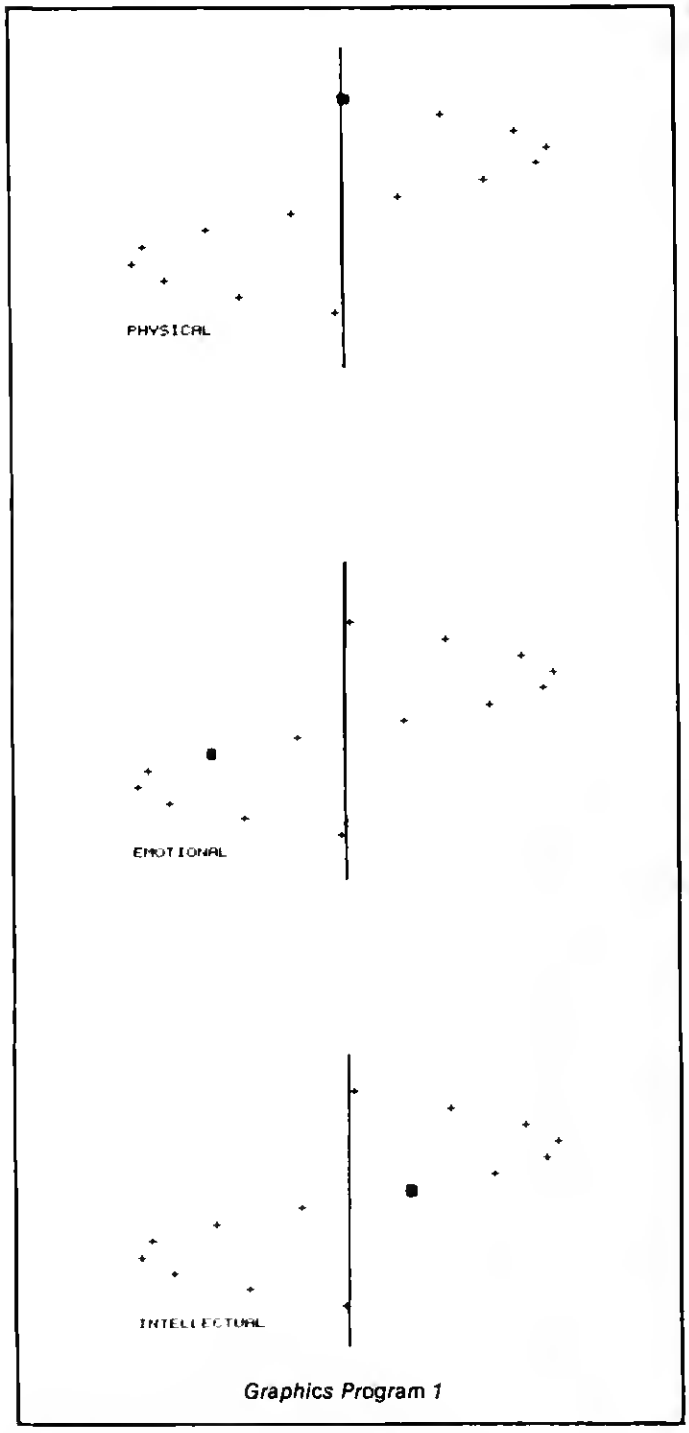
The intellectual cycle effects our understanding, adaptability, logic, wit, judgement and concentration. The plus period, lasting 16½ days, is the best time for study, planning, examinations and decisions.

The minus period, also lasting 16½ days, is a period in which we are apt to be lacking in good judgement. It is a good time for gathering data or for jobs that require repetition.

Critical days are those during which a cycle crosses the centerline (vertical in the program) in an upswing (to the right) or a downswing (to the left). The term critical is relative. It is a period of change where our system is in a stete of flux.

Statistical research for over 30 years seems to disclose that during these critical days, especially the physical and emotional ones, we are more accident prone, lacking in coordination, judgement and alertness. During an emotionally critical day, one is apt to make e slip of the tongue, irresponsible utterances, quarrel or have disputes. An intellectually critical day might cause failure of memory or mistakes. Critical days occur on our computer program where the blinking signal is on the vertical centerline of the curve or very close to it.

Photo 1: Here, note that the marker is on the line: a critical day.



Graphics Program 1

The Program

Whether or not you believe in biorhythms, you can still have a lot of fun running this program. Lines 2 through 11 are standard print statements and self explanatory. In lines 15 and 80 dates are input. It is important that they be entered correctly—for instance May 8, 1979 would be entered 5,8,1979. The full year must be written, not merely 79, and commas must be

Photos 2 and 3: The rectangle, which is the marker, flashes on the screen and shows the location of that day's biorhythm—emotional (Photo 2), intellectual (Photo 3).

used for the Level II BASIC on the TRS-80.

Statements 20 through 70 are a calendar routine to find the number of days to day of birth. Lines 20, 50 and 60 take care of the months of January and February which are considered the 13th and 14th months of the previous year in the formula.

Statements 90 thru 140 also make up a calendar routine for finding the number of days to present date as entered.

to today's date as entered. Statement 160 calculates today's position in the physical cycle of 23 days. This is a fraction of the complete cycle of 23 days. A subroutine could be used for steps 20 to 60 and 90 to 130, however I did not feel that it was worth the bother.

Statements 180 through 200 draw the sine curve. For ease in plotting, this curve is drawn on its side. In line 180 the curve is initiated from zero to 2π, which is one complete sine wave since the curve a slightly different form.

Lines 210 through 240 draw the zero line of the curve; this utilizes the TRS-80's graphic ability.

Lines 250, 260, 170 and 280 plot today's physical location on the curve, while lines 265 and 285 establish a timer loop. This timer loop makes the marker blink.

Line 280 can be omitted fixing the marker on the screen for the length of the timer loop.

In line 190 one is added to the sine to make it positive. It is also multiplied by 20 in order to print the curve more clearly on the screen. (I found these the best values for a nice looking sine wave.)

In statements 250, 390 and 590 the X value (horizontal) of the respective biorhythm is calculated. Statements 260, 400 and 600 give the Y value (vertical).

The constants .273182, .2244 and .1904 represent the values of 2π divided by 23, 28 and 33, the biorhythm periods of the physical, emotional and intellectual cycles.

The position of the marker for subsequent dates can be estimated or plotted day by day by inputting the appropriate dates. For those who do not have graphics capability on their computer or anyone wishing to substitute a numerical value for the graph there is an alternate program. It is much simpler. Steps 2 through 11 are omitted, substituting any printed messages that the programmer wishes. Lines 15 through 150 are retained as shown in the main program listing. Lines 160 onwards could be changed as in Example 1.

The same sort of thing can be done for the emotional and intellectual cycles changing the cycle interval in statement 160 appropriately to 28 or 33. ■

```
160  P = DT/23 - INT(DT/23)
170  BP = SIN(P * 2 * 3.14159265)
----then something like this could be added----
175  PRINT
180  PRINT
190  PRINT " ON A SCALE OF MINUS ONE TO PLUS ONE"
200  PRINT " THE NUMERICAL VALUE OF YOUR PHYSICAL BIORHYTHM IS "; BP
210  PRINT
```

Add this low cost interface and you'll be able to turn on any memory location you want for control or monitor use.

I/O Ports Plus

Brian A. Herron
67-3691 Albion Rd.
Ottawa Ontario
Canada K1T1P2

By adding the low-cost interface described in this article, you will be able to "PEEK" and "POKE" your way around in the real world outside your TRS-80 cabinet.

To begin, let's examine Level II BASIC's PEEK and POKE commands. The POKE X,Y command will store the value Y into memory location X, where Y is a decimal number between 0 and 255 representing an 8-bit binary byte and X is the decimal value of any writable memory location. Conversely, the PEEK(X) command will return the value (0 to 255), which is read from memory location X(decimal).

This all seems useful, but what shall I PEEK at and where will I POKE? Well, let us add some input/output ports in memory address space, then we will be able to send and receive bytes of data via these new ports, or registers.

The Interface

Fig. 1. shows the connections necessary to add an Intel 8255 programmable peripheral interface IC to the TRS-80 bus. On the right side are our 24 input/output lines, each capable of sourcing 1 mA of current at 1.5 volts (TTL compatible) and on the left side are the TRS-80 bus connections and pin function names.

The 7404 hex inverter and the 7430 8-input NAND gate are required to properly decode the memory addresses where our I/O registers will reside. These address decoders will allow the 8255 to be selected (via $\overline{CS}$) only when data is read or written at addresses 12288-12291 (3000-3003 hex). These locations were chosen because they were in an unused area of memory just above the last Level II ROM and well below the keyboard scan RAM area. It also seemed very improbable that Radio Shack peripherals would ever use these locations.

It should be noted at this time that with Level II BASIC installed in your TRS-80, the internal 5-volt supply is not available to the user, so an external 5-volt source capable of providing at least 500 mA of current will have to be made available. All other required lines are available at the 40-pin edge connector located inside the rear access door of the CPU/keyboard housing.

The 8255 has three modes of operation that may be selected under program control: mode 0—basic input/output, mode 1—strobed input/output with interrupt support. For our interface we will confine our thoughts to mode 0 only. Further details concerning the 8255 PPI can be found in the manufacturers' data handbooks.

Port A and B are 8-bit ports and port C is split into two 4-bit ports. Port C has the additional feature of offering bit set/reset capability. All outputs are latched, while the inputs are not. The port function configuration depends upon which

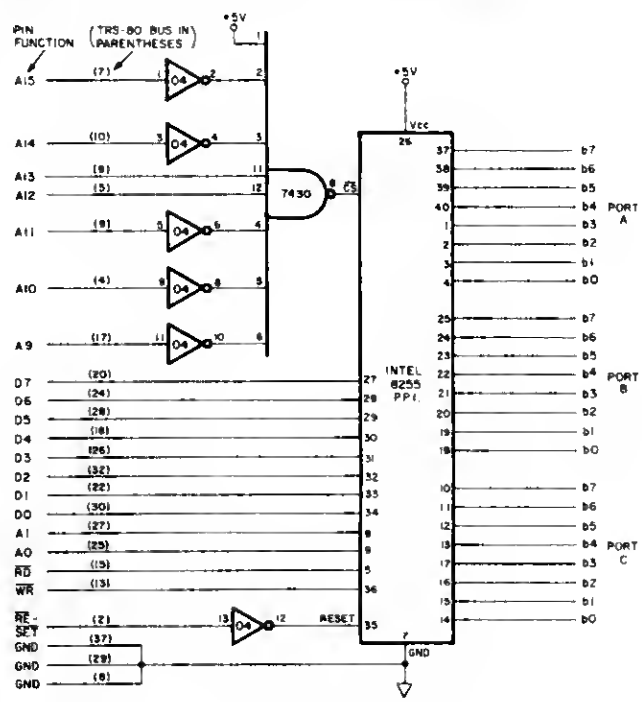


Fig. 1. Schematic of 24 I/O line interface.

```
10  POKE 12291,153      ;REM A-IN B-OUT C-IN CONTROL WORD
20  I = PEEK(12288)     ;REM SET I = VALUE AT PORT A
30  J = PEEK(12290)     ;REM SET J = VALUE AT PORT C
40  IF I = J THEN POKE 12289,240   ;REM 240 = 11110000 BINARY
50  GOTO 10
```

Sample BASIC program to demonstrate the 8255 interface.

Control Word (Decimal)	Port A (b0-b7)	Port C Upper (b4-b7)	Port B (b0-b7)	Port C Lower (b0-b3)
128	OUT	OUT	OUT	OUT
129	OUT	OUT	OUT	IN
130	OUT	OUT	IN	OUT
131	OUT	OUT	IN	IN
136	OUT	IN	OUT	OUT
137	OUT	IN	OUT	IN
138	OUT	IN	IN	OUT
139	OUT	IN	IN	IN
144	IN	OUT	OUT	OUT
145	IN	OUT	OUT	IN
146	IN	OUT	IN	OUT
147	IN	OUT	IN	IN
152	IN	IN	OUT	OUT
153	IN	IN	OUT	IN
154	IN	IN	IN	OUT
155	IN	IN	IN	IN

Table 1. Input/output mode 0 control word chart.

control word is POKEd into address 12291 (3003 hex). Sixteen different combinations of input and/or output are listed in Table 1.

For example let's POKE 12291,137. Port A (address 12288) is an 8-bit output port, port B (address 12289) is another 8-bit output port and port C (address 12290) is an 8-bit input port.

If port C were configured to be an output port (i.e., POKE 12291,144), you could turn on or turn off any of the individual bits of port C by POKE 12290,Z, where Z is a word from Table 2 defining which bit is to be acted upon. This becomes handy for controlling custom peripherals that require strobing or mode-setting bits of data. For example, consider the challenge of controlling eight railroad model switches or turning house and yard lights on and off.

Let's try a BASIC program example where we will look at ports A and C, and if they are equal we will turn on bits 4, 5, 6, 7 or port B. The sample program will keep looping and show a binary 11110000 on port B whenever ports A and C are equal.

One final warning: If after adding this interface you still cannot think of anything to control and/or monitor, then your TRS-80 may suffer from "terminal" boredom!■

Control Word (Decimal)	Port C Bit Set or Reset	
0	b0 RESET	
1		b0 SET
2	b1 RESET	
3		b1 SET
4	b2 RESET	
5		b2 SET
6	b3 RESET	
7		b3 SET
8	b4 RESET	
9		b4 SET
10	b5 RESET	
11		b5 SET
12	b6 RESET	
13		b6 SET
14	b7 RESET	
15		b7 SET

Table 2. Port C bit set/reset control word chart.

A Level I program to help you document your programs.

Screen Editor

William L. Colsher
4328 Nutmeg Lane, Apt. 111
Lisle IL 60532

What, you might ask, is a screen editor and why should I want one for my TRS-80? Well, first I'll tell you why you should want one so you'll be motivated to read the rest of this article and get the program running on your system. (And by then you'll know what one is.)

There is one absolutely horrendous problem with programming small computers in BASIC (or in any other non-compiled language, for that matter). Even with a small program, once you put in all the REMarks and start in on the user instructions, you begin to find that space is limited. In fact, I'm sure you've seen programs that have no built-in instructions (or even REMarks) for this very reason. This problem is so completely universal that most of us don't even think of it as a problem.

The obvious solution to the problem of providing good user instructions in a limited amount of space is to segregate them from the main program in a module of their own. As soon as you do that, however, you begin to find that writing instructions that are going to be displayed on a video screen can be a real pain in BASIC.

The first thing you realize is that typing P." nine zillion times can be boring. Then, when you RUN the program to take a look at the instructions, they zip by too fast to see. So you have to go back and stick in either an INPUT statement every 15 lines (on a TRS-80) or some sort of timing loop to slow things down. Finally, if your program requires a lot of instructional material (as is the case with some of the sophisticated simulation games), you can still run out of memory and have to add one or more additional instruction modules.

Nondestructive Cursor

The solution to all this miserable fooling around with PRINTs and two or three extra programs is (you guessed it) a screen editor. A screen editor is actually a simple sort of text editor. Since all we're concerned with is one screen full of information, the editor doesn't have to be smart. All that's needed is a nondestructive cursor to get the information where we want it and some mechanism for saving the information for future use. Because the requirements are simple, a screen editor is simple to use—no messing around with PRINTs and wait loops.

If you think about it for a few minutes, you will find that the nondestructive cursor is the major part of writing a screen editor. On a TRS-80 (or any other system where the cursor is an automatic part of keyboard input), there is an additional problem. The automatic cursor fouls up the nondestructive goal. It has to go.

The way to get rid of an unwanted cursor is simple: Just

1. Load the screen editor tape with CLOAD. Since this is a machine-language tape, the asterisk won't blink very much.

2. When the monitor screen clears, you're ready to enter a screen of information. There are four special keys used to move the cursor around the screen:
UP—the "up arrow"— —(hold down the shift key) moves the cursor *up* one line.
DOWN—the "down arrow"—↓—(hold down the shift key) moves the cursor *down* one line.
RIGHT—the "right arrow"—→—(hold down the shift key) moves the cursor *right* one space.
LEFT—the "left" or "back arrow"—←—(hold down the shift key) moves the cursor back one space.

3. When the screen is just the way you want it, ready your tape recorder and hit the "@" key. If this is the *last* screen of data, hit the "L" key. If there are more screens to enter, hit the "M" key. When the screen clears again you're ready to edit.

4. When you have recorded the last screen of data, press the RESET button on the back of the TRS-80. Now CLOAD the BASIC program you wrote the instruction screens for and make sure it loaded correctly. Then CSAVE it onto the end of the instruction screen tape.

5. To use your composite tape, load it into your recorder, press play, type CLOAD. A screen full of information will be displayed as fast as it can be loaded from tape. When you want to continue to the next screen, hit enter and it will be loaded. After you hit ENTER on the last screen, the READY message will appear and you can then type CLOAD again to load the BASIC program.

Screen editor user's guide.

delude the routine that handles it. All we have to do is see to it that the offending routine thinks the video display memory is someplace it isn't. . . preferably out of the way where the cursor, trailing its cloud of characters, can't do any harm to our program or our neatly edited display. This turns out to be quite simple on the TRS-80. There is a location in RAM where the TRS-80's built-in routines store the current location of that nasty cursor. That location is at the two bytes beginning at 4068H. As is usually the case with an address, it is stored backward, high-order byte second.

The only safe place to make the cursor think the video memory is turns out to be the ROM area. Anyplace else might affect the program. In ROM though, the cursor can gaily write characters all day and not do the slightest amount of damage. To make sure it stays "down on the ROM" all we have to do is stick a zero at location 4069H. If we do that every time we input a character from the keyboard we'll always be safe. There are probably more elegant ways to pull this off, but it only takes two lines of code (starting at 4411H in the program listing) and doesn't materially affect the speed of the program.

Now that we have the built-in cursor permanently out of the way, we need to take a look at how to make our cursor nondestructive. That means that when we back it up, for example, the cursor should just slide over the characters already typed. Of course, at any given location, the cursor character itself is what's on the screen. The cursor character we're using is the underscore _. When we move the cursor to another location, the character that was there before the cursor reappears.

This is easy to implement. Every time a cursor commend is input from the keyboard, we immediately put the old character back where the cursor was. Then we generate a new location for the cursor (depending on the command entered), save the character at that location and then write in the cursor. The four lines that make up the "save" part of this routine begin at location 4416H in the program.

Cursor Commands

Cursor commands are special characters that are used to tell the program what direction to move the cursor. A quick glance at the TRS-80 keyboard reveals that there are four arrow keys. Since each arrow points in one of the four directions (right, left, up and down), we want to move the cursor so we don't have to use up any of the other characters to get our work done.

While I was working on this program, it occurred to me that some people might want to use the arrows in their display. To make this possible, I decided to use the shifted arrows for the cursor commands. This also has the advantage of keeping the cursor from flying all over the screen if you accidentally hit one of the control keys.

In addition to moving the cursor around, we also need some way to tell the program that we've finished with the screen and to save it on tape. I decided to use the "at" symbol (@). (The reason for this is just that I've never used it for anything else and decided it was about time.)

Because there are only five commands to be processed, we can use the brute-force method instead of using branch tables, etc. Our method consists of a series of compares and jumps. If a comparison is true, then we jump off to handle the command. The 15 lines starting at location 441DH contain this series of jumps. Incidentally, if you don't like the characters I picked out for control purposes, just substitute your own choices in the compare (CP) statements.

Program listing

```
                        *-----------------------------------------*
                        *                                         *
                        *            TRS-80 SCREEN EDITOR         *
                        *                                         *
                        *  AUTHOR - WILLIAM L. COLSHER            *
                        *  SYSTEM - TRS-80, 4K, LEVEL I           *
                        *                                         *
                        *  THIS SIMPLE SCREEN EDITOR WAS WRITTEN TO *
                        *  SIMPLIFY DEVELOPMENT OF INSTRUCTIONAL  *
                        *  MATERIAL TO ACCOMPANY BASIC LANGUAGE   *
                        *  PROGRAMS.                              *
                        *                                         *
                        *-----------------------------------------*
                        *
                        *  THE FOLLOWING SEGMENT OF CODE CLEARS THE
                        *  MONITOR SCREEN IN PREPARATION FOR EDITING
                        *
4400  21003C    START     LD   HL,3C00H      START OF VIDEO MEMORY
4403  11013C              LD   DE,3C01H
4406  010004              LD   BC,0400H      BYTE COUNT
4409  3E20                LD   A,20H         PUT A BLANK IN FIRST SPOT
440B  77                  LD   (HL),A        ..
440C  EDB0                LDIR               CLEAR THE SCREEN TO BLANKS
                        *
                        *  THE NEXT SECTION OF CODE PROCESSES THE INPUT.
                        *  IF A CONTROL CHARACTER IS INPUT, IT IS HANDED
                        *  OVER TO THE APPROPRIATE ROUTINE.  ORDINARY
                        *  CHARACTERS ARE PLACED ON THE SCREEN AND THE
                        *  CURSOR LOCATION INCREMENTED BY ONE.
                        *
                        *  CONTROL CHARACTERS:
                        *
                        *  5B - UP ARROW - MOVES CURSOR UP
                        *  5C - DOWN ARROW - MOVES CURSOR DOWN
                        *  5D - LEFT (OR BACK) ARROW - MOVES CURSOR LEFT ONE
                        *  5E - RIGHT ARROW - MOVES CURSOR RIGHTONE SPACE
                        *  40 - THE @ SYMBOL - GIVES CONTROL TO SAVE ROUTINE.
                        *
440E  21003C              LD   HL,3C00H      INITIAL CURSOR LOCATION
4411  3E00      EDIT      LD   A,00H         KILL RADIO SHACK CURSOR
4413  326940              LD   (4069H),A     . .
4416  7E                  LD   A,(HL)        GET CURRENT CHARACTER
4417  32004C              LD   (4C00),A      AND SAVE IT
441A  3E5F                LD   A,5FH         PUT IN OUR CURSOR
441C  77                  LD   (HL),A        . .
441D  CD400B      WAIT    CALL KBDIN         READ A KEYBOARD CHARACTER
4420  CA1D44              JP   Z,WAIT        . .
4423  FE5B                CP   5BH           IF UP ARROW THEN
4425  CA4644              JP   Z,UP          MOVE CURSOR UP
4428  FE5C                CP   5CH           IF DOWN ARROW THEN
442A  CA5144              JP   Z,DOWN        MOVE CURSOR DOWN
442D  FE5D                CP   5DH           IF LEFT ARROW THEN
442F  CA5C44              JP   Z,LEFT        MOVE CURSOR LEFT
4432  FE5E                CP   5E            IF RIGHT ARROW THEN
4434  CA6344              JP   Z,RIGHT       MOVE THE CURSOR RIGHT
4437  FE40                CP   40H           IF @ THEN
4439  CA6A44              JP   Z,SAVE        SAVE THE SCREEN ON TAPE
443C  77                  LD   (HL),A        OTHERWISE PUT IT ON THE SCREEN
443D  23                  INC  HL            MOVE OUR CURSOR OVER ONE
443E  C31144              JP   EDIT          AND CONTINUE EDITING
```

```
                        *
                        *  THIS IS THE "RESTORE" ROUTINE.  IT IS CALLED
                        *  EVERYTIME A CURSOR COMMAND IS PROCESSED TO
                        *  RESTORE THE CHARACTER THAT WAS 'UNDER' THE CURSOR
                        *  BEFORE IT MOVED.
                        *
4441    3A004C          RESTORE LD   A,(4C00H)      GET THE CHARACTER
4444    77                      LD   (HL),A         PUT IT BACK
4445    C9                      RET                 RETURN TO CALLER
                        *
                        *  THE FOLLOWING CODE HANDLES ALL MOVEMENT OF THE CURSOR
                        *  INITIATED BY A CONTROL CHARACTER.
                        *
4446    CD4144          UP      CALL RESTORE        PUT BACK THE OLD CHARACTER
4449    114000                  LD   DE,0040H       64
444C    ED52                    SUB  HL,DE          (HL) - 64
444E    C31144                  JP   EDIT
                        *
4451    CD4144          DOWN    CALL RESTORE        PUT BACK THE OLD CHARACTER
4454    114000                  LD   DE,0040H       64
4457    ED5A                    ADD  HL,DE          (HL) + 64
4459    C31144                  JP   EDIT
                        *
445C    CD4144          LEFT    CALL RESTORE        PUT BACK THE OLD CHARACTER
445F    2B                      DEC  HL             (HL) - 1
4460    C31144                  JP   EDIT
                        *
4463    CD4144          RIGHT   CALL RESTORE        PUT BACK THE OLD CHARACTER
4466    23                      INC  HL             (HL) + 1
4467    C31144                  JP   EDIT
                        *
                        *  THE FOLLOWING CODE HANDLES THE SAVING OF COMPLETED
                        *  SCREENS ON TAPE.  NOT THE SLIGHTLY DIFFERENT
                        *  PROCEDURE FOR THE LAST SCREEN SAVED.
                        *
446A    CD8B44          SAVE    CALL READER         SET UP PROGRAM TO GO WITH TAPE
446D    000000                  NOP,NOP,NOP         NO OP'S IN PLACE OF TEST CODE
4470    000000                  NOP,NOP,NOP
4473    000000                  NOP,NOP,NOP
4476    000000                  NOP,NOP,NOP
4479    00                      NOP
447A    CD890F                  CALL CTON           TURN ON TAPE RECORDER
447D    21003C                  LD   HL,3C00H       POINT TO START OF SAVED AREA
4481    110042                  LD   DE,4200H       POINT TO END OF SAVED AREA
4484    CD4B0F                  CALL CSAVE0         SAVE IT
4487    C30044                  JP   START          AND BEGIN AGAIN
                        *
                        *  THE FOLLOWING CODE HANDLES THE GENERATION OF THE
                        *  SMALL PROGRAM WHICH WILL GO ON TAPE WITH THE
                        *  SCREENS.  ALL SCREENS BUT THE LAST HAVE A ROUTINE
                        *  WHICH WAITS FOR AN 'ENTER'.  THEY THEN CLEAR THE
                        *  MONITOR AND READ IN THE NEXT SCREEN.  THE LAST SCREEN
                        *  WAITS FOR AN 'ENTER' AND THEN CLEARS THE MONITOR
                        *  FINALLY JUMPING TO LOCATION 0000H TO INITIALIZE BASIC
                        *
448B    CD400B          READER  CALL KBDIN          READ A CHARACTER FROM THE KEYBOARD
448E    FE4C                    CP   4CH            IS IT AN L?
4490    CAAD44                  JP   Z,LAST         YES, DO SPECIAL STUFF
4493    FE4D                    CP   4DH            IS IT AN M?
4495    CA9B44                  JP   Z,MORE         YES, SET UP STANDARD PROGRAM
                        *
449B    011900          MORE    LD   BC,0019H       # BYTES IN PROGRAM
449E    210040                  LD   HL,4000H       START OF PROGRAM
44A1    22F841                  LD   (41F8H),HL     TELL TRS-80 ABOUT IT
44A4    21BC44                  LD   HL,44BCH       POINT TO COPY OF PROGRAM
44A7    110040                  LD   DE,4000H       DESTINATION OF PROGRAM.
44AA    EDB0                    LDIR                MOVE IT IN
44AC    C9                      RET                 GO BACK AND SAVE THE STUFF
                        *
44AD    21C300          LAST    LD   HL,00C3H       SET UP A JP TO 0000H
44B0    22D244                  LD   44D2H,HL       STORE FIRST 2 BYTES OF JP
44B3    210000                  LD   HL,0000H       GET 2 MORE BYTES OF 0'S
44B6    22D444                  LD   44D4H,HL       STORE SECOND PAIR OF BYTES
44B9    C39B44                  JP   MORE           AND GO DO THE REST.
                        *
                        *  THIS SECTION OF CODE IS NOT EXECUTED IN THIS PROGRAM.
                        *  IT IS THE PROGRAM THAT IS STORED WITH THE SCREENS TO
                        *  CHAIN THEM TOGETHER.
                        *
44BC    CD400B                  CALL KBDIN          READ KEYBOARD
44BF    FE0D                    CP   0DH            ENTER?
44C1    C20040                  JP   NZ,4000H       NOT YET
44C4    21003C                  LD   HL,3C00H       SET UP TO CLEAR SCREEN
44C7    11013C                  LD   DE,3C01H       . .
44CA    01FF03                  LD   BC,03FFH       . .
44CD    3E20                    LD   A,20H          GET A BLANK
44CF    77                      LD   (HL),A         STICK IT IN
44D0    EDB0                    LDIR                CLEAR THE SCREEN TO BLANKS
44D2    CDF40E                  CALL CLOAD0         LOAD NEXT SCREEN
```

Actually, processing the cursor commands is simple too. The only special thing we have to do is make sure that the old character is pulled out of storage and put back where it belongs when the cursor is moved. A little routine called RESTORE (only three lines long beginning at 4441_{16}) handles this. After that's done, we can generate a new cursor address by adding or subtracting 1 or 64, depending on the direction we want to move.

Handling the "save" command is trickier. In fact, about a third of this program is used to do it. The reason for this is simple. We want to make our instruction screens easy to use. The most obvious way to simplify things is to eliminate any need for the user to do anything but read the information and tell the program when he's done. So, each screen full of information has to have a little program along with it that waits for the user to hit ENTER. When he does, the routine clears the screen and starts loading the next instruction screen.

Because BASIC programs load differently, we have to do something different with the last screen of information. All this involves is changing that little program so that when the user hits ENTER instead of loading another instruction screen, the computer displays the READY message.

Applications

The screen editor can be used for other things besides making up instruction screens for BASIC programs. With another program to help it, you can use it as a sign generator. In a 4K TRS-80, you can store three different pages of information. The additional program consists of a time delay routine and a couple of statements to move in a new display. This can be used for an automated bulletin board at club meetings, or a cable TV station could use it for its local news and weather channel. Most places tie up a perfectly good wall and camera on this right now. You can probably think of other uses. ■

Add your own error messages to Level II.

Extra Errors

Charles Moses
New England Digital Corp.
PO Box 305
Norwich VT 05055

One feature of TRSDOS is that Disk BASIC describes non-disk errors rather than give a two-character abbreviation as does Level II. Though Level II offers a number of error commands (ERR, ERL, etc.), making it easy to write an error-trapping routine, I wanted one that would do more.

A closer inspection of the operating system revealed that for syntax errors the Editor is automatically invoked. Looking over the list of Error Codes in the Level II manual, I decided there were other errors which might be corrected quickly by using the Editor. Consequently, I needed three things from my routine: a more detailed explanation of errors; flexibility to decide which errors I wanted to handle as special cases and those that Level II could handle in the normal way; a way of invoking the Editor for immediate repairs like syntax errors.

PRINT Statements

The first is satisfied by using PRINT statements containing a one line description of the error. The second can be satisfied in a variety of ways, such as in Example 1.

The IF THEN statement traps those errors that you want the operating system to handle. You could also send all those errors to the same line number in the

```
10 ON ERROR GOTO 1000
.
.
.
*1000 IF ERR = 10 OR ERR = 12 OR
ERR = 14 THEN ON ERROR GOTO
0
**1010 ON ERR/2 + 1 GOTO 1020,
1000, 1040,,,,,,1050,,,,,,,,1060
1020 PRINT "NEXT WITHOUT FOR
IN LINE"; ERL
.
.
.
2000 END
*ERR = (Error Code – 1)*2 or ERR/
2 + 1 = Error Code. ON ERROR
GOTO 0 is the way you get out of
the error-trapping routine and let
Level II handle the error.
**Level II will pass over the com-
mas as though they were line
numbers. If ERR/2 + 1 = 's a value
for which there is no line number
you will get a "?UL ERROR IN
1010" error!
```

Example 1

Example 2

```
2 REM EXAMPLE OF ERROR
HANDLING ROUTINE
4 REM SYNTAX ERRORS WILL
LIST THE LINE AND GO TO EDIT
6 REM ALL OTHERS REQUIRE
THE USE OF THE EDIT COM-
MAND 'L'
8 REM TO LIST THE LINE AS IN
THE NORMAL EDITOR (ERL = ER-
ROR LINR NUMBER
100 ON ERROR GOTO 1000
110 CLS: DEFINT I
120 FOR I = 1 TO 10
130 PRINT I;
140 NEXT I
150 END
*1000 ON ERR/2 + 1 GOTO 1010,
1020, 1030, 1200, 1200, 1200, 1040
(23 codes)
1010 PRINT:PRINT "NEXT
WITHOUT FOR IN"; ERL: EDIT.
**1020 PRINT:PRINT "SYNTAX
ERROR IN"; ERL: LIST.
1030 PRINT:PRINT "RETURN
WITHOUT GOSUB"; ERL: EDIT
1040 PRINT:PRINT "OUT OF
MEMORY IN"; ERL: EDIT
***1200 ON ERROR GOTO 0
2000 END
*Line 1040 is in the seventh posi-
tion so it is the seventh error code.
Give different line numbers for the
errors you want to handle; give
1200 for all the others
** List. accomplishes what the
Editor would normally do but with
the added twist that the line is
already listed for you. Using EDIT.
for syntax errors will work except
that the command is stacked
twice and after the program runs
ok, the Editor will try to EDIT the
line an error occurred on again,
after the READY appears.
***The normal error and/or Editor
will be invoked when this line is
read, automatically.
```

ON ERR/2 + 1 GOTO statement, containing the single statement ON ERROR GOTO 0.

Number 3, the most difficult, was satisfied using another curious ability of Level II: A command, like RUN or LIST can be executed from a BASIC statement like this:

```
10 PRINT "HELLO"
20 RUN 10
```

For the third we will use "EDIT." (and "LIST.", explained later) which is supposed to edit the current line. But, when used in an error-trapping routine, its current line is that in which the error occurred! We can invoke the Editor by placing the command EDIT in the program inside the error routine, as shown in Example 2.

If the program is run and there are no errors the following

should appear:

```
1 2 3 4 5 6 7 8 9 10
READY
>
```

If there is a syntax error, like making line 130 read; 130 RINT I;, this appears:

```
SYNTAX ERROR IN 130
130 RINT I;
READY
130
```

You can then edit using all the normal commands and RUN the program again. If there was an error in line 140—140 NEXT W, you would see:

```
NEXT WITHOUT FOR IN 140
140                (Type an 'L')
140 NEXT W
140                (Use Editor normally)
```

If two lines are added to the program; 115 DATA 1,2,3 and 125 READ X, there would be a number four error code which sends the error to 1200 and out to the normal error messages and response.

```
?OD ERROR IN 125
READY
>
```

Every possible error can be tested by using the ERROR command/statement. By listing a line in the program like 105 ERROR 1 (or 2, or 20 . . .), your error trapping code can be tested directly, without having to create a specific error.

Conclusion

Using this routine has saved me thumbing through the Level II manual to find the error I've created and saved time by being able to go directly into the Editor.

The idea of the routine was to trap certain errors which could be corrected by producing the error line, but obviously, some errors cannot be handled that way. For example, a UE Unprintable Error or an L3 Disk Basic Only, must be corrected by the operating system.

Also, a LIST ?-? works well for isolating a cluster of related lines, for instance a loop, as in the example program LIST 120-140. Even without the long statements describing the errors a LIST ?-? could be used to look at a number of different sections of the program at once. The idea is flexibility. ■

How to make sure you have the right people, in the right place—at the right time.

Duty Roster

Dick M. Straw
891 Michigan
Pasadena CA 91104

If one of your tasks at the office, in your church or in your club involves putting people into schedules on a fair, random basis, Duty Roster will help.

Many organizations can use an effective but simple way of sharing the work load. Assigning church ushers and acolytes to their weekly duties is a good example. Here, the object is to try to use everyone about equally often and to spread them out as much as possible. In Toastmasters' Clubs and others, various tasks are taken by different members each week. Making up a month's schedule can be a problem.

Maybe you want a method to distribute the necessary evening duties in your store as equitably as possible, and a fixed rotation is undesirable—after all, who wants to be stuck working Friday night *every* week? The Duty Roster program will do all of these tasks.

Coding

Written for the TRS-80 in Level II BASIC, this program uses about 8K of storage, but the total will really depend on how long a list of names is included in it. It will assign every individual on the list to any job available until the list is exhausted, or it can restrict assignments to some jobs to persons known to be qualified to do them. For example, if the duties of the chairman of a meeting are rotated, but it is desired to limit the post to members with more experience, a second, coded option should be chosen.

In the standard coded option, individuals with higher codes can be assigned to lower-level jobs, but higher-level tasks will not be given to persons whose codes are too low. If there are really separate categories, so that some persons should be assigned only to one group of tasks and others to a different group of tasks, a simple change in two program lines will accomplish it.

The acolyte-and-usher problem falls into the last basket, probably. But a little imagination will point out some other occasions to use the program. For example, proper coding will allow random assignment of partners for the weekly bridge club meeting. Should you want to be sure your daily menu would include each of the major food groups, you could rotate the acceptable choices randomly to get a varied fare!

As currently implemented, four categories, coded 1 to 4, are allowed. Some improvement in efficiency results if fewer can be used in practice, but more can also be added. Naturally, as individuals gain experience, their code levels can be increased, and job-level codes can also be changed as appropriate.

How the Program Works

The program uses information stored in DATA statements at its end. Line 4990 carries three parameters: the number of jobs, the number of persons and the option code (zero if restricted assignments, nine if not). The data format is different for each.

Lines 5000 and up carry the job names and their associated codes in pairs (for example, 5000 DATA "CHAIRMAN",4, "HOST",3,). Higher-code tasks should be listed first for best efficiency to avoid assigning all the high-code personnel to lower-level jobs before they can be used in the higher ones.

From line 5100 onward are the

✓ Reader Service—see page 147

persons' names in the same format. The parameters and the data must all agree with one another in order to make the program go. As you well know, of course, the DATA statements can be anywhere you like so long as they are in the right order. The codes are omitted for option "nine."

After the preliminaries are out of the way and you tell the program to proceed, the data are read (lines 800-900) and checked (the subroutine beginning at line 4300). To begin with, there must be at least as many persons to assign as there are weekly tasks. That is not usually a problem. This check will also trap you, in most cases, if you reverse the order of the first two parameters, although it will not give an appropriate diagnostic. It then counts how many jobs there are at each level (if coded) and how many persons on the list are qualified to fill each. This is reported out. If the number of qualified persons is too small for a single weekly roster, the program will stop at the job level where this is detected.

Then the shuffling begins. The random list is constructed in array K such that each index number is used only once. This is done by the subroutine beginning at line 4100. This segment also zeroes the counters and resets any negative job codes in the person-list.

Assignments are taken care of in the main program. For unrestricted assignments, the simple routine begins at line 1100. It reads the random K array and places the individual's index numbers into the assignment array, K2. When the list is exhausted before the end of the run, it calls the shuffling routine and continues on. This sometimes causes the same person to be given two jobs on one day, but if the list of people is long enough compared to the job list, that should be rare. You can eyeball a switch or run it all over again.

Restricted postings are made beginning at line 1200. It operates basically the same way, but checks to make sure the person assigned has a high enough rating for each job. When an assignment is made,

Duty Roster program.

```
1 REM DUTY ROSTER PROGRAM  (C) DICK STRAW, 1979
50 CLEAR 3000: CLS
60 ON ERROR GOTO 4500
70 DEFINT H-M: DEFSTR G,P
100 PRINT CHR$(23)
110 PRINT@334,"DUTY ROSTER"
120 FORI=1TO500:K=1:NEXTI
130 CLS:PRINT"THIS PROGRAM WILL ASSIGN JOBS TO INDIVIDUALS ON"
135 PRINT"A RANDOM OR STRATIFIED RANDOM BASIS "
140 PRINT:INPUT"DO YOU WANT A BRIEF PROGRAM DESCRIPTION",P9
150 READ N,M,L0
160 DIM G(N),L(N),P(M),L2(M),K(M),K2(N,4),H(4)
170 REM G=JOBS,L=JOB CODES,P=PERSONS,L2=PERSON CODES
180 REM K=RANDOM LIST,K2=ASSIGNMENTS,H=COUNTERS
200 FORI=1TON:L(I)=0
210 FORJ=1TO4:K2(I,J)=0
220 NEXT J,I
230 FORI=1TON:L2(I)=0:K(I)=0:NEXTI
300 IF ASC(P9)=89 GOSUB 400
310 GOTO 700
400 REM PROGRAM DESCRIPTION
410 PRINT:PRINT"THE PROGRAM PROVIDES DUTY ROSTERS FOR FOUR PERIODS (WEEKS),"
420 PRINT"BASED ON A RANDOMIZED LIST.  IF THERE ARE NOT ENOUGH PERSONS"
430 PRINT"LEFT FOR THE NEXT ROSTER, THE LIST IS RE-SHUFFLED."
435 INPUT"     FOR MORE, PRESS ENTER";P9
440 CLS:PRINT"IN ONE OPTION, EVERY PERSON IS ELIGIBLE FOR ALL JOBS."
450 PRINT"IN THE OTHER, JOBS AND PERSONS ARE CODED TO RESTRICT"
460 PRINT"ASSIGNMENT TO SOME JOBS TO MORE QUALIFIED PERSONS."
470 PRINT"A PERSON CANNOT BE ASSIGNED A HIGHER-CODE JOB, BUT"
480 PRINT"MAY BE GIVEN A LOWER-CODE JOB.  TWO SIMPLE MODIFICATIONS"
490 PRINT"WILL ASSIGN PERSONS ONLY WITHIN THEIR CATEGORIES."
500 PRINT"ASSIGNED CODES ARE EASILY MODIFIED.  JOBS OR PERSONS"
510 PRINT"LIMITED TO ONE ASSIGNMENT SHOULD BE OMITTED FROM LISTS."
520 PRINT:PRINT"PROGRAM PARAMETERS ARE IN DATA LINE 4990."
530 PRINT"JOB NAMES AND CODES ARE IN DATA LINES STARTING AT 5000."
540 PRINT"PERSONS AND CODES ARE IN LINES 5100 AND UP."
550 PRINT"SAMPLE DATA FOR TRIAL RUN ARE INCLUDED.  FOLLOW THE"
560 PRINT"EXAMPLES TO ENTER YOUR OWN DATA.  OMIT NUMERIC CODES"
570 PRINT"FOR NO-RESTRICTION OPTION.  CODES 1 TO 4 ALLOWED."
580 RETURN
700 PRINTTAB(5);:INPUT"ARE YOU READY TO PROCEED (Y/N)",P9
710 IF ASC(P9)=89 GOTO 800
720 CLS:PRINT:PRINT"ALL RIGHT, WHEN YOU HAVE ENTERED THE DATA, JUST"
730 PRINT"'RUN' THE PROGRAM AGAIN.  REMEMBER THAT YOU CAN 'CSAVE'"
740 PRINT"THE PROGRAM WITH THE DATA TO USE AGAIN LATER."
760 STOP
800 CLS:PRINT"READING DATA"
810 IF L0=0 GOTO 850
820 FOR I=1 TO N
822 READ G(I)
824 NEXT I
830 FOR I=1 TO M
832 READ P(I)
834 NEXT I
840 GOTO 890
850 FOR I=1 TO N
852 READ G(I),L(I)
854 NEXT I
860 FOR I=1 TO M
862 READ P(I),L2(I)
864 NEXT I
890 GOSUB 4300
900 GOSUB 4100
910 PRINT"WORKING ON THE ASSIGNMENTS"
1000 IF L0=0 GOTO 1200
1090 REM UNRESTRICTED ASSIGNMENTS
1100 FOR I9=1 TO 4
1110 FOR J9=1 TO N
1120 K2(J9,I9)=K(H(0))
1130 H(0)=H(0)+1:  IF H(0) > N GOSUB 4100
1140 NEXT J9
1150 NEXT I9
1160 GOTO 2000
1190 REM RESTRICTED ASSIGNMENTS
1200 FOR I9=1 TO 4
1205 IF N > M-H(0) THEN GOSUB 4100
1210 FOR J9=1 TO N
1220 L1=L(J9)
1230 K1= K(H(L1))
1240 IF L2(K1) >= L1 GOTO 1290
1245 REM FOR ONLY-IN-CATEGORY, CHANGE >= TO = IN L  1240
1250 H(L1)=H(L1)+1  IF H(L1) <= M GOTO 1230
1260 H(L1)=1
1270 FOR I=1TOM
1272 IF L2(I)= -L1 THEN LET L2(I)= -L2(I)
1274 NEXTI
1280 GOTO 1230
1290 K2(J9,I9)=K1
1300 L2(K1)= -L2(K1)  H(0)=H(0)+1
1320 NEXT J9
1330 NEXT I9
2000 FOR I9 = 1 TO 4
2010 CLS PRINT"HERE ARE THE ASSIGNMENTS FOR WEEK",I9
2020 PRINT TAB(7);"JOB",TAB(35);"PERSON"
2050 FOR J9=1 TO N
2060 PRINT G(J9), TAB(30), P(K2(J9,I9))
2070 NEXT J9
2100 INPUT"WHEN READY PRESS ENTER",P9
2110 NEXT I9
3000 CLS PRINTCHR$(23)
3010 PRINT@330,"HOPE IT WAS HELPFUL"
3020 PRINT@470,"THE END"
3030 FORI=1TO500 K=1 NEXTI
3040 CLS
4000 END
4100 RANDOM PRINT"SHUFFLING"
4110 FOR I=1 TO N
4120 K5=RND(M)
4130  FOR J=1 TO I
4140  IF K(J)=K5 GOTO 4120
4150  NEXT J
4160 K(I)=K5
4170 NEXT I
4180 FOR I=0TO4  N(I)=1  NEXT I
4190 FOR I=1TON  L2(I)=ABS(L2(I)):  NEXT I
4200 RETURN
4300 REM CHECKING ROUTINE
4310 PRINT"CHECKING DATA"
4320 PRINT"YOU HAVE",M,"PERSONS FOR",N,"JOBS "
4330 IF M >= N GOTO 4370
4340 PRINT"WE CANNOT PROCEED UNLESS THERE IS AT LEAST ONE PERSON"
4350 PRINT"FOR EACH JOB.  SUGGEST YOU REDEFINE JOBS " STOP
4370 FOR I=0TO4 H(I)=0 NEXTI
4380 IF L0=9 GOTO 4470
4390 FOR I=1 TO N
4395 IF L(I)<1 OR L(I)>4 PRINT"JOB CODE OUT OF RANGE " STOP
4400 H(L(I)) = H(L(I))+1
4405 NEXT I
4410 FOR I=4 TO 1 STEP -1
4415 FOR J=1 TO M
4420 IF L2(J)<1 OR L2(J)>4 PRINT"PERSON CODE OUT OF RANGE" STOP
4425 IF L2(J)=I THEN H(0)=H(0)+1
4430 NEXT J
4435 PRINTTAB(5),H(0);"PERSONS CAN TAKE JOBS OF LEVEL";I
4440 IF H(0) >= H(I) GOTO 4450
4445 PRINT"NOT ENOUGH PERSONS QUALIFIED FOR THESE JOBS."  STOP
4450 REM IF ONLY-IN-CATEGORY WANTED, INSERT H(0)=0 HERE
4460 NEXT I
4470 RETURN
4500 REM ERROR HANDLING
4510 IF ERR=24 OR ERR=2 PRINT"YOUR DATA ARE NOT ENTERED PROPERLY" STOP
4520 IF ERR=6 PRINT"NOT ENOUGH DATA -- CHECK AND RESTART" STOP
4540 IF ERR=8 THEN P9="N" RESUME
4560 ON ERROR GOTO 0
4910 REM DATA LINE 4990 HAS N=# OF JOBS, M=# OF PERSONS, AND
4920 REM CODE FOR OPTION, 0 = RESTRICTED, 9 = NOT RESTRICTED
4990 DATA 7,20,0
4995 REM JOB DATA FROM LINE 5000   PUT HIGHEST CODE JOBS FIRST
5000 DATA "CHAIRMAN",3,"SECRETARY",2,"SERGEANT-AT-ARMS",1
5010 DATA"LEADER 1",1,"LEADER 2",1,"LEADER 3",1,"LEADER 4",1
5095 REM PERSON DATA LINES 5100 ON
5100 DATA "JONES",3,"SMITH",1,"STONE",2,"HARDY",2,"WALLACE",1
5110 DATA "EVANS",3,"ROGERS",3,"ADAMS",1,"WHITE",1,"CASEY",1
5120 DATA"QUINCY",2,"CROSS",1,"WILLIAMS",3,"RISK",2,"OWENS",1
5130 DATA"JACKSON",1,"JOHNSON",3,"TAYLOR",2,"TANDY",1
5140 DATA "BAX",2
```

the individual's job-level code is set negative so he will always be "unqualified" for subsequent jobs until a re-sort and reset occurs. If not enough persons remain unassigned for a whole day's roster, the list will be reshuffled.

Customizing the Program

Because there may be fewer highly qualified persons than are needed for the full four-week array of rosters, an internal adjustment of the counters and codes allows those premium individuals to recycle more often than the rest. The counters in array H, by the way, allow the search to begin where it left off last time.

Then everything is printed out a week at a time by the segment beginning at line 2000. The job and people's names are accessed with the index numbers that have been shuffled around up to this point.

In order to assign persons within a single category only, two changes are needed. One is to change the "greater than or equal to" symbols in line 1240 to "equals." The counter in the checking subroutine should be set back to zero at the end of each loop by inserting line 4450 H(0) = 0.

Since out-of-range codes are trapped, the number of valid codes can only be increased by changing the dimensions of array H (line 160) and the indices for loops that use the code levels, at lines 4180, 4395, 4410 and 4420. The number of weeks allowed can be increased by changing the second dimension of array K2 at line 160 and the statements that use it—210, 1100, 1200 and 2000. All other arrays and loops are set by the input parameters.

Although the program is written in Level II BASIC on the TRS-80, there is little unique in it. The special features that are used are either nonessential (such as the 32-character displays at the beginning and end) or are easily written around. Two string arrays are declared by a DEFSTR statement at line 70. Conversion might require seeking out the few places the string arrays are referenced (in the dimension statement at line 160, the reading statements between 800 and 900 and the printout routine beginning at line 2000) and changing the Gs and Ps to G$ and P$. The guts of the program are standard BASIC.

Now, when you rotate the kids among the dish-washing, yard-work, car-washing and dog-walking chores, at least it can be purely random and fair. ■

Want to see how much you spend each month? This program plots time versus rate.

Graph Plotter

Scott King
7905 59th Ave. N.
New Hope, MN 55428

Have you ever had to create a need to put in front of your solution? That was my problem. After I spent the money for a Level II TRS-80 (16K), my wife said, "Kind of expensive for a Star Trek game, isn't it?" I had to accept the reason I bought a computer: it was really neat.

For those of you who join me in the middle-to-low income bracket, "really neat" is hard to justify to the wife and the budget. So I set out to create a need for the computer that I already had.

One of the returns on my endeavor was this program to display my gas, electric, water, etc., bills on a time graph so that I might compare the bad months to the good ones, check to see if my energy-conservation measures were doing any good and plan ahead for bills that fluctuate throughout the season (e.g., my gas bill).

Program listing.

```
10 REM ***** TIME/RATE DISPLAY PROGRAM ****
20 REM ***** BY SCOTT KING    2/16/79  ****
30 REM ***** TRS-00 12K 0ASIC          ****
40 REM ************************************
45 CLS
50 PRINT"    TIME/RATE DISPLAY PPOGRAM   "
55 PRINT" PRESS ENTER TO BEGIN ";:INPUT AA
90 CLEAR 1000
100 DEFSTR A,0,C,D
110 DIM A(75)
120 N=1
130 INPUT"ENTER X AXIS LABEL ";B
140 INPUT"ENTER Y AXIS LABEL ";C
141 IF LEN(C)>15 THEN PRINT" THIS LABEL IS LIMITED TO 15 CHARACTERS":GOTO 140
145 INPUT"ENTER FIRST X AXIS INCREMENT  I E. JAN' MONDAY' ETC ";D
148 INPUT " WILL THERE BE MORE THAN 16 ENTRIES MADE",AA
149 IF AA<>"NO" THEN Z1=1
150 INPUT" IS THERE A TAPE TO LOAD";AA
160 IF AA="NO" THEN 200
170 INPUT" LOAD THE TAPE AND PRESS ENTER";AA
171 INPUT#-1,B
172 PRINT"X- AXIS =";B
173 INPUT#-1,C
174 PRINT"Y- AXIS =",C
175 INPUT#-1,D
176 PRINT"FIRST X INCREMENT",D
177 INPUT#-1,Z1
178 IF Z1=1 PPINT" HIGH DENSITY DISPLAY
180 INPUT#-1,A(N)
185 PRINT N,A(N),
190 IF A(N)="END" THEN GOTO 200
195 N=N+1 : GOTO 180
200 INPUT " ARE THERE ANY ENTRIES TO BE MADE ";AA
210 IF AA="NO" THEN 240
220 PRINTN INPUT "ENTER DATA AMOUNT";A(N)
230 N=N+1  GOTO 200
240 A(N)="END"
250 INPUT"DO YOU WISH TO CHANGE ANY ENTRIES";AA
251 IF AA="NO" THEN 260
252 INPUT"ENTER NUMBER TO BE CHANGED";N
253 INPUT"ENTER NEW VALUE";A(N)
260 CLS
270 M=1
271 PRINT "X= ";B
272 PRINT "Y= ";C
273 PRINT "X STARTS AT ";D
280 PRINT M,A(M),:IF M<=N THEN M=M+1:GOTO 280
290 INPUT" DO YOU WISH TO SAVE THIS DATA ON TAPE";AA
300 IF AA="NO" THEN 350
305 INPUT"SET UP A TAPE FOR RECORDING AND PRESS ENTER";AA
310 M=1
311 PRINT#-1,B
312 PRINT#-1,C
313 PRINT#-1,D
314 PRINT#-1,Z1
320 PRINT# -1,A(M)
330 M=M+1:IF M<=N GOTO 320
```

The Program

This program was written on a TRS-80 Level II, but with minor changes, it will easily run on a Level I. It is designed to plot any two sets of numbers on an x-y type graph. The x (horizontal) scale has two different resolutions, 16 and 75 points. If you are working with the high-density scale (75 points), you can "blow up" a section of it for a better look.

The y-axis (vertical) is self adjusting. Its lowest resolution is 0 to 20, but it will automatically adjust to the largest number to be displayed.

Operation

The program starts by asking you to label the x-axis (for this example we will use months, but it could be days, weeks, hours or even, in some cases, people, money, cars . . . anything, as long as it doesn't exceed 75).

Next, enter the y-axis label, which may not exceed 15 characters. We will use kilowatt-hours (kWhr) for this example. So this graph will be a plot of kWhr used over a period of months.

The next entry to be made is the starting point of the x-axis; we will begin ours at January 1979.

Now the program will ask if

there is an old tape to load. If this is the first time you have run this, then enter no. But if you have data, load the tape and dump it into memory.

If there are any manual entries to be made, such as a new bill arrival, then enter them at this juncture. Next, you will have the opportunity to change any entries previously made.

After all of this has taken place, the computer will display the current data and ask if you wish to save it on tape (which you should do). When this is done, the computer will draw the graph, go through the y-axis automatic scaling and plot the data on the display. The graph will stay until you press ENTER, at which point it will ask if you would like an expanded view of any section.

Conclusion

I have found this program useful in keeping track of my bills, but it could easily be used to plot a curve for any situation where you have both time and rate numbers. ■

```
350 REM *** PLOT DATA *****
351 CLS
360 S=LEN(C)
370 W=1:T=0
380 PRINT@T,MID$(C,W,1)
390 W=W+1: IF W>S THEN 500
400 T=T+64:GOTO 390
500  PRINT @ 970,0," STARTING WITH ";D;
510 FOR X=5 TO 110
520 Y=40 SET(X,Y):NEXT
530 FOR Y=0 TO 43
540 X=5:SET(X,Y)
541 X=110:SET(X,Y)
542 NEXT
543 IF Z1=1 THEN FOR X=10TO110 STEP 2:Y=41:SET (X,Y):NEXT
546 IF Z1=1 THEN 550
547 G=099:F=1
548 PRINT@G,F:F=F+1:G=G+3:IF G<945 THEN 540
550 GOSUB 650
555 GOSUB 700
560 X=10
565 N=1
570 FOR Y= 40 TO (40-(VAL(A(N))/K)) STEP -1
575 SET(X,Y):NEXT
577 IF Z1=1 N=N+1:X=X+2:IF A(N)="END" THEN 600
578 IF Z1=1 THEN 585
580 N=N+1:X=X+6: IF A(N)="END" THEN 600
585 IF (Z1=0)AND(N>16) THEN 600
586 GOTO 570
600 INPUT A$
620 INPUT " DO YOU WISH TO SEE A BLOWUP OF ANY ONE SECTION";AA
621 IF AA="NO" THEN RUN
622 INPUT" 16 PLACES WILL BE DISPLAYED... ENTER FIRST POINT TO BE SHOWN";M
623 FOR V=1TO10:A(V)=A(M):IF A(M)="END"THEN A(V)="END":GOTO 625
624 M=M+1:NEXT
625 A(V)="END":Z1=0
626 GOTO 350
650 REM **** SIZING ROUTINE *****
655 K= 5:M=1
660 IF A(M)="END" RETURN
662 IF M>74 THEN RETURN
665 Z=VAL(A(M))
670 X=Z/K
675 IF X> 40 THEN K=K+.5:GOTO 660
680 M=M+1 GOTO 660
700 L=57 H=40
710 PRINT @ L,INT(H+K);
720 L=L+64 H=H-3
730 IF H>=0 THEN 710
740 RETURN
```

Keep all your wires out of sight, but under control with this cabinet idea.

Box It In

John Zalnerunas
3034 W. Columbus Ave.
Chicago IL 60652

Ever since my TRS-80 was delivered, the problem of strung-out wire, cables and components taking up too much of my already limited space kept cropping up. This is, no doubt, a problem with many computers, particularly home-brew systems, but it should not be a problem with a fully assembled store-bought unit. I was determined to correct this before attempting any serious programming.

A console "walnut veneer" custom TRS-80 is shown in the photo. It is portable, turns on with one switch and plugs in with one cord. It even has a tape-recorder control switch to bypass the computer.

The "power on" and "tape control" additions were taken from previous issues of *Kilobaud*. (see Lien and Waterman's articles, "Cassette Recorder Disaster: Ground Loops," p.110, May 1978, and "Turn It Off," p. 114, April 1978)

The visible part of the console is constructed of walnut veneered, shelf boards. These boards have a brown-grained appearance, a good match for the silver gray and black of the TRS-80.

Computing is more organized and professional with a homemade console for your TRS-80 system. The author's son Paul illustrates how relaxing programming can be.

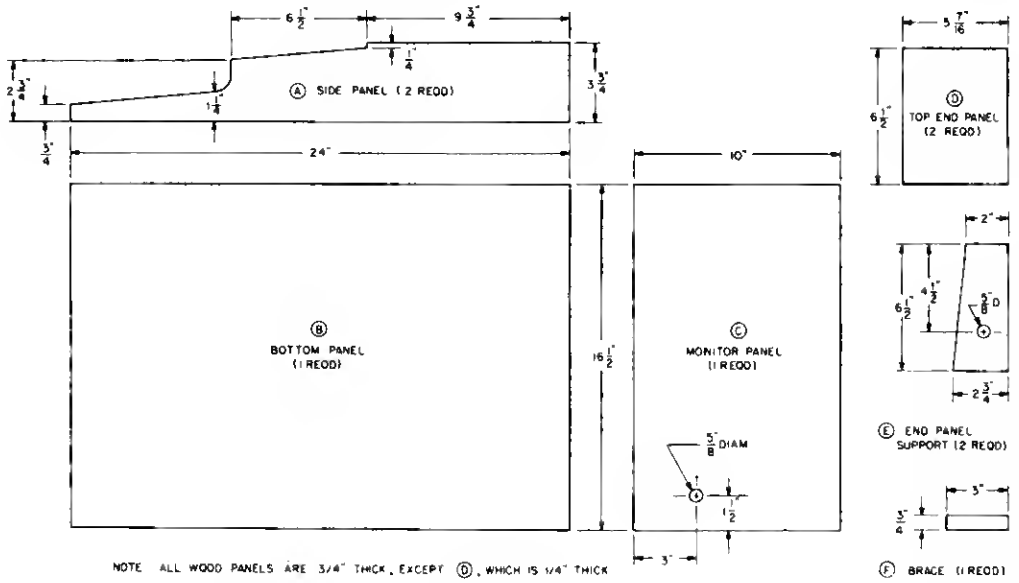


Fig. 1.

I used a 3/4 inch plywood panel for the full one-piece bottom support and wood-stained all visible edges to match the walnut veneer color.

You can use pine boards in place of the walnut boards, if desired. These can be cut into panels and covered with a wood-grained "contact paper." Later you can add some metal trim for a professional look.

Construction

Start by cutting up all panels to exact dimensions as shown in Fig. 1. Drill holes and make up the required amount of each panel as indicated. Mounting screw holes are not dimen-

sioned but can be positioned approximately as in Fig. 2. Use a #8 or #10 brass wood screw, 1½ inches long, for all mountings.

First assemble both (A) panels to the bottom plywood board. Pre-drill holes, apply white glue to mounting edge and screw panels together. You can recess the screws so they can be filled in with wood putty for a neater look.

Mount the top monitor panel (C) next by applying white glue and screwing this panel to both side panels. I even used white glue on all screw threads for extra holding power, since the walnut panels were made of particle board.

Now glue in the two small (E) supports and the center brace (F) to the bottom plywood. Use a two-part epoxy glue so mounting screws will not be required.

A simple bracket for the mini-jack can be made as shown in Fig. 2 and mounted on the left side panel. A spring plate is mounted on the center front edge of the plywood board. Press this plate edge into the slotted seam of the keyboard to hold it in place. Use two of them side by side if needed. I used a plate made from a piece of .020 phosphorous bronze.

Final Assembly

Place your tape recorder over the center brace (F) and push it under the monitor panel. The recorder should fit snugly, right up to its counter switch. (Add shims to brace, if required.)

Wire an SPST switch to the left end (D) panel for recorder control, as shown in Fig. 3. Now epoxy this panel in place. Epoxy the right (D) panel and hold down in place with some weights; screws are not required.

Place your computer keyboard over the plywood panel. Connect and route all cables as shown. Push the keyboard forward to check the fit. You probably will have to carve out a small part of the right side panel (A) to make a better fit for the far right cable plug on the keyboard. When a good fit is obtained, push the keyboard fully forward and attach the locking spring plate. This spring plate allows the computer keyboard to be moved back slightly to get at the reset and off/on switches when required.

If you decide to mount the monitor permanently to the top panel (C), use #8 wood screws—1½ inches long with large washers and mounted from inside of monitor case (through slots), one in each corner.

With the console completed, programming is now more enjoyable. ■

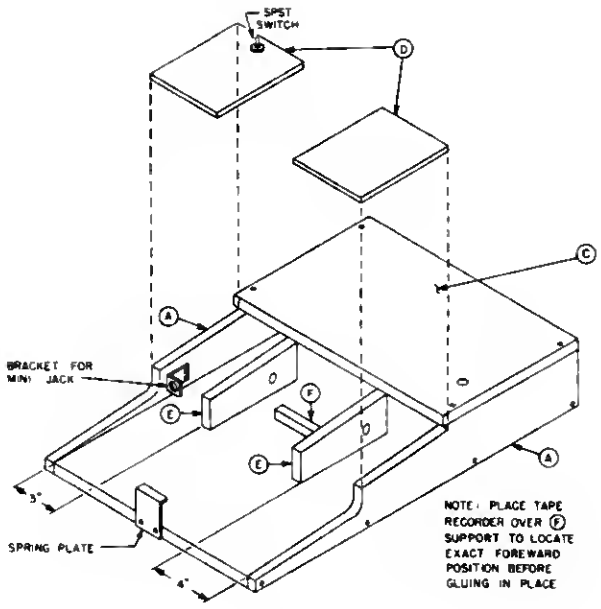


Fig. 2.

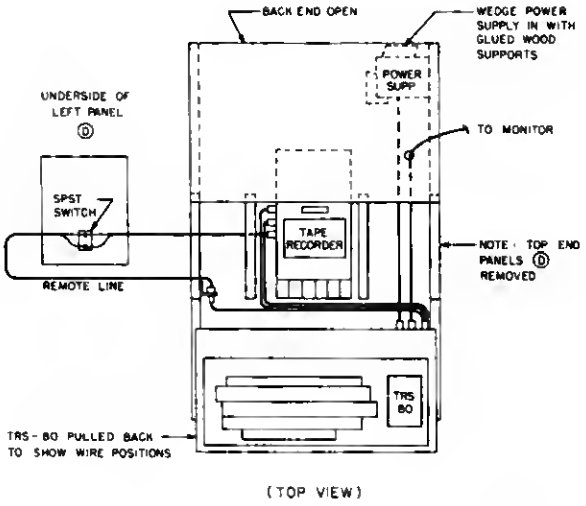


Fig. 3.

Apparat software allows old programs in new BASIC.

One into Two

Sherman R. Wantz
424 NW Lakeview Dr.
Sebring FL 33870

If you have been giving serious thought to modifying your TRS-80 Level II BASIC system so that it will accept all of those Level I BASIC programs you have saved on cassette tape, wait. You don't have to void your Radio Shack warranty by opening your Level II machine's case, cutting circuit board traces, installing a Level I ROM and adding several resistors and a switch. There's a simpler way.

Apparat, Inc., of Denver, Colorado, offers a software program on cassette tape that will make your Level II BASIC system act as though it were a Level I BASIC machine. The price of the "Level I In Level II RAM" program cassette is a modest $14.95, plus $2 for hendling and shipping.

If you have very many Level I BASIC program cassettes on hand that you haven't been able to convert for use in your Level II system, the price of the program is quite reasonable. I was never able to get Radio Shack's "Program Conversion, Level I to Level II" cassette to work satisfactorily.

Apparat's program is written in machine language and occupies about 4.3K bytes of memory space at the upper end of your Level II 16K byte memory bank. That leaves almost 12K bytes of memory available for use by any Level I program you want to RUN. There are some—but not many—programs written in Level I BASIC that require that much memory.

The program cassette supplied by Apparat comes without any supporting documentation. However, almost all you have to know to use the program is written in just seven words and symbols that appear on a label attached to the cassette's plastic shipping box.

After you turn "on" your Level II BASIC system, you type SYSTEM on your keyboard and press the Enter key. The "*?-" prompt will appear on your monitor's screen to indicate that the computer is ready to accept a machine-language program. You then place the "Level I In Level II RAM" program cassette in your tape recorder and engage the recorder's Play lever.

Type LEVEL I and press the Enter key. This will activate your recorder's motor and will start loading the program into memory. If your recorder's volume control has been adjusted correctly, you should see the usual double asterisks (one of which blinks) appear in the upper-right corner of your video monitor's screen.

If you fail to obtain the double asterisk/blinking indication during loading, rewind the cassette tape, reset your recorder's volume control, press the reset button inside the left rear cover of your computer's keyboard assembly and repeat the loading procedure described above.

After the "Level I In Level II RAM" program has been transferred to your computer's memory, the tape recorder's motor will stop turning and the "*?-" prompt will again appear on your monitor's screen. Type "/" and press the Enter key. The computer will clear the monitor's screen, will print READY and will display the familiar ">-" BASIC prompt. From that point on, your Level II machine will perform as though it were operating under control of BASIC contained in a Level I ROM.

Rewind the Apparat tape and remove the cassette from your recorder. Insert in the recorder the Level I program tape you wish to use, activate the recorder's Play lever, type CLOAD on your keyboard and press the Enter key.

As your Level I program is transferred from tape to memory, you will notice that the double asterisk now appears in the upper *left* corner of your monitor's screen. When your Level I program has been loaded into memory, the recorder's motor will stop and the ">-" prompt will reappear. Now, all you have to do to use the program in memory is type RUN and press the ENTER key.

Operating In Level I

I have been particularly pleased to find that I do not have to readjust my recorder's volume setting after I load the "Level I In Level II RAM" program and then CLOAD my Level I program. I think that condition is attributed to just plain luck on my part. But after having fiddled so long with my recorder's volume control to load my Level II program tapes made on other recorders, it is refreshing to recall how uncomplicated it used to be to load tapes a year ago when my system was using Level I BASIC.

The computer's response to the LIST command fills the monitor's screen with program lines. You use the "↑" (up-arrow) key to step through successive program lines. You can "dump" (record) your program to cassette tape using the CSAVE command. The format of the recorded program will be Level I and the data transfer rate will be 250 baud.

Even the error messages are pure Level I, i.e., "What," "How" and "Sorry." If you have forgotten what these error messages mean, you had best dig out that dusty "User's Manual For Level I" and refresh your memory.

Of course, since your system is now operating in Level I BASIC, you do not have access to the convenient editing functions that Level II BASIC provides. Nor will you be able to use the LLIST and LPRINT commands that Level II provides for printout of your programs.

You will be pleased to learn that when you use the NEW command, the Level I BASIC program in the computer is erased, but the "Level I In Level II RAM" program remains. Therefore, you will be able to load successive Level I BASIC programs into your computer without having to reload Apparat's program.

However, if one of your programs' "hangs up" during loading and you are forced to activate the computer's reset push-

button switch to regain control, you will probably have to reload the "Level I In Level II RAM" program before you can load another Level I BASIC program. As soon as you turn your computer off, the "Level I In Level II RAM" program disappears and your system reverts to its Level II BASIC configuration.

Limitations

My experience with Apparat's program has uncovered one or two minor problems that I attribute to my having loaded the program at too low a volume setting on my recorder.

One of my programs contained a line that used A$ and B$ separated by a colon. The B$ string was continually truncated (shortened) even though it consisted of less than the allowed 16 characters. I moved B$ to a separate program line and had no more problems.

Another problem I experienced involved my use of the ON N GO TO statement. In one portion of my program, the computer would not respond to the shorthand dialect ON N G. —,—. However, in several other parts of the same program, the computer did accept the shorthand statement without default.

Since these problems have not occurred in any of a number of other Level I programs I have RUN using the Apparat program, I assume that the problems alluded to above were self-induced and were not the fault of the conversion program.

I have heard that the "Level I In Level II RAM" program will not accept programs written in machine language. Since Apparat did not furnish documentation with its program tape to warn against attempting to use the program with machine code, perhaps it will perform properly in exactly the same manner that a Level I BASIC program in ROM would operate. Since I have no machine-code programs in my library, I could not test the conversion program to validate the claim that it is incompetible with machine language.

I'm very pleased to have the capability that Apparat's program provides. The price was right and the program's operation has been better than satisfactory. My only fear is: Murphy's Law is bound to come into play and, somehow, I'll inadvertently erase my program tape. But that's just the way my luck runs.

For those of you who might want to order the Level I In Level II RAM program cassette write to: Apparat, Inc., 6000 E. Evans Ave., Bldg #2, Denver CO 80222. ■

The Radio Shack board lets your 80 do the talking—to other computers.

RS 232

Rogar L. Hicks
5534 Woodberry Circle
Meriatta GA 30067

Communicetions has come to the TRS-80, and, at $99, it may be the biggest bargain sinca Laval II BASIC. I am referring to the RS-232C board (shown at right).

Even if you are not interested in communicatlons, raad on! The RS-232 board has a variaty of usas thet do not involva communications in tha usual sansa of the word.

Most of us are familiar with a parallal intarfaca, which movas all aight bits simultanaously ovar aight saparata linas. A serial intarface handlas tha same eight bits, but ona at e time ovar a singla lina. Tha Electronics Industrias Association (EIA) has established a standerd for serial interfaces ratarrad to as RS-232C. Whila this approach claarly lands itsalf to telecommunications, there are othar usas.

Tha RS-232 is tha only faatura I know of offarad by Radio Shack that glvas the usar a standard way of connecting a variety of paripheral davicas from othar manufacturars. In my case, I was able to implement a low-cost hard-copy lina printar into my system.

Implamantation

The board in the accompanying photo installs under the larga accass cover in your existing expansion interface and comas with a ribbon cable, instruction manual and machine-languaga program that allows your TRS-80 to emulate a standard asynchronous terminal (providad a modam, acoustical coupler or direct connact is available). The board plugs into a 42-pin connactor that is part of the expansion interface. Soma early production varsions have the connector already in placa, but you will probably have to take your expansion interface to your nearast Radio Shack sarvice centar for installation (no extra charge).

The aarly version of the instruction manual has a number of errors (howevar, the quality of the rast of the feature is excellent). Table 5 on page 16 erroneously shows 130 baud as tha result of loading 5H into the BRG (baud rate generator). A 5H will, in raality, yiald 300 baud.

The DIP switches are labeled on the board (S1-S8), but nowhere is the user told in which direction to push a DIP switch to opan or close the circuit. To open, push aach switch in the direction of the UART (large IC chip). The remaining errors ara minor (easy to figura out), and all remaining implemantation instructions are clear.

Line Printar Application

Tha ramaindar of this article dascribes tha programming considerations in using the RS-232 intarface to add a line printar to a 16K system (I axpact this to be the most common noncommunications application). Rafarring to your Laval II BASIC manual, tha trick is getting your RS-232/printer to raspond to LPRINT and LLIST commands. Baginning on page 26, the RS-232 manual describes the procedure.

The line printer driver program is in ROM and is dasignad for a printer connectad to the standard parallel interface. This

```
5 POKE 16421,2:POKE 16422,183:POKE 16423,127
10 FOR X=32695 TO 32767:READ Y:POKE X,Y:NEXT
15 END
20 DATA 229,197,245,58,255,127,254,1,40,32,62,1,50,255,127
25 DATA 211,232,219,233,230,248,246,4,50,254,127,211,234
30 DATA 219,233,230,7,33,246,127,6,0,79,9,126,211,233,241
35 DATA 193,225,219,234,203,119,40,250,121,211,235,254,13
40 DATA 32,4,14,10,24,239,201,34,68,85,102,119,170,204,238,0,0
```

Listing 1.

```
10 POKE 16413,2:POKE 16414,183:POKE 16415,127
5000 POKE 16413,7:POKE 16414,88:POKE 16415,4
```

Listing 2.

driver program is, however, a subroutine that is called each time a character is to be printed. This call address is in RAM and can be modified (POKE) to execute a user-supplied driver stored in RAM. The assembly-language listing of this driver begins on page 27 of the manual, and a BASIC program to POKE the machine-language code into upper RAM is on page 29. My only problem here was wasting 256 bytes for a driver program that only needs 73.

Listing 1 is a modified version of the program shown on page 29, except that it only occupies the last 73 RAM locations. On power up, enter a memory size of 32694 (instead of 32511 for the Radio Shack version). Load and run the BASIC program; you can now enter NEW and proceed with any program or activity using LPRINT and LLIST at will. Execution of NEW will not erase the driver program beginning in location 32695 (it will, in fact, remain until you power-down).

If you are like me, you have at least one large program with a lot of PRINT statements that you will now want to change to LPRINT. Wait, there is an easier way! It so happens that the driver program for the video display is also a ROM-resident subroutine called by an address stored in RAM.

Referring to Listing 2, if you add line 10 to the beginning of your program, all subsequent PRINT statements will go to the RS-232/printer instead of the video display. Be sure to execute line 5000 before stopping, or the video's output will continue to go to the printer (if you get an unanticipated break that prevents this, simply enter RUN5000). Needless to say, if you are redirecting video output to the printer, you cannot use PRINT @ or video graphics.

One Final Problem

The software we have covered assumes that at 300 baud, for example, each character can be decoded and printed in the time it takes the RS-232 to handle one byte (approximately 34 milliseconds). This is a valid assumption unless the character decodes to a carriage return (in which event, the driver program automatically inserts a line feed as the next character). Depending on the type of printer involved, the next one (or more) printable character(s) may arrive before the printer is ready (and be lost).

If your printer manual does not cover this subject, it can be corrected (if encountered) in one of two ways:

1. After each LPRINT that completes a line, add

LPRINT STRING$(5, CHR$(00));

This transmits five (subject to change) nonprintable characters, which gives the printer an extra (5 × 34) 170 milliseconds to get ready.

2. Execute a FOR-NEXT loop to stall the program, for example:

FOR X=1 TO 100:NEXT X

With a little luck, you will not encounter the problem except at high baud rates (1200+).■

✓ Reader Service—see page 147

Soundex is used to class similar sounding names together, for easier genealogical research.

Soundex Codes

Robert A. Hodge
417 Pelham Street
Fredericksburg, VA 22401

I have been involved in genealogical research for a number of years, and occasionally have found it necessary to use the records housed in the National Archives. There, a number of records, particularly the latest available census records, are indexed according to an elaborate coding system called Soundex.

Developed to group similar sounding names together regardless of spelling, the system uses the first letter of a name for alphabetical filing and converts all other letters into numbers as follows:

- 1 is assigned to b,f,p,v;
- 2 is assigned to c,g,j,k,q,s,x,z;
- 3 is assigned to d,t;
- 4 is assigned to l;
- 5 is assigned to m,n;
- 6 is assigned to r;
- no value is assigned to a,e,h,i,o,u,w,y (normally);

Complications sometimes arise because only one of any two consecutive equivalents are recorded (including, in this case, the first letter, normally not coded at all). The final result must be adjusted to consist of one letter and a 3-digit number, extra digits being discarded; absent ones are recorded as zeros.

I have found it nearly impossible, especially under a time pressure, to code a list of names without making errors. Because of this, I try to prepare all codes to be searched well in advance of attempting to use the records.

Working with my TRS-80, I have developed a program which accurately produces the Soundex code for any name entered.■

```
100 CLS
105 PRINT TAB(20)"THE SOUNDEX CODE"
110 PRINT:PRINT"A NUMBER  OF RECORDS IN THE NATIONAL ARCHIVES HAVE BEEN INDEXED
    ACCORDING  TO  A  SYSTEM  ALLOWING  SIMILAR  SOUNDING  NAMES TO BE GROUPED
    TOGETHER REGARDLESS OF SPELLING.  THIS   SYSTEM   IS TERMED THE 'SOUNDEX SY
    STEM'."
115 PRINT:PRINT"IF ONE IS GOING TO THE ARCHIVES,   IT WOULD BE HELPFUL AND TIME-
    SAVING TO HAVE THE SOUNDEX    CODES   ALREADY  AVAILABLE FOR THE SURNAMES (
    LAST NAMES) ONE WILL BE SEEKING."
120 PRINT:PRINT"I AM PROGRAMMED TO DETERMINE THE SOUNDEX CODE FOR YOU":PRINT:INP
    UT"TAP ENTER WHEN READY";A
125 DIMA$(25),B$(25),C(25)
130  CLS
135 PRINT"TYPE THE SURNAME (LAST NAME) TO BE CODED AND PRESS 'ENTER'":INPUT"";A$
140 A=LEN(A$)
145 C(1)=ASC(LEFT$(A$,1))
150 FOR  B=2 TO A
155 B$(B)=MID$(A$,B,1)
160 NEXT B
165 FOR B=2 TO A
170 C(B)=ASC(B$(B))
175 NEXT B
180 FOR B=1 TO A
185 IF C(B)=66 OR C(B)=70 OR C(B)=80 OR C(B)=86 THEN C(B)=1
190 IF C(B)=67 OR C(B)=71 OR C(B)=74 OR C(B)=75 THEN C(B)=2
195 IF C(B)=81 OR C(B)=83 OR C(B)=88 OR C(B)=90 THEN C(B)=2
200 IF C(B)=68 OR C(B)=84 THEN  C(B)=3
205 IF C(B)=77 OR C(B)=78 THEN  C(B)=5
210 IF C(B)=76 THEN C(B)=4
215 IF C(B)=82 THEN C(B)=6
220 IF C(B)>6 THEN C(B)=0
225 IF C(B-1)=C(B) THEN C(B)=0
230 NEXT B
235 CLS:X=0
240 PRINT"THE SOUNDEX CODE FOR '"A$"' IS:"
245 PRINT:PRINT:PRINT LEFT$(A$,1),
250 FOR B=2 TO A
255 IF C(B)=0 GOTO 275
260 PRINT C(B),
265 X=X+1
270 IF X=3 GOTO 295
275 NEXT B
280 FOR B=1 TO 3-X
285 PRINT "0",
290  NEXT B
295 PRINT:INPUT"IF YOU  HAVE ANOTHER NAME, PRESS 'ENTER' OTHERWISE TYPE 'NO'";A$
300 IF A$="NO" GOTO 305ELSE   130
305 CLS:END
```

Program Listing.

HOBBY WORLD ELECTRONICS

America's Largest Mail-Order Computer Store

CALL TOLL FREE:
(800) 423-5387 USA
IN CALIF:
(800) 382-3651
LOCAL & OUTSIDE USA
(213) 886-9200

MATCHLESS SYSTEMS

TRS-80 MINIDISK DRIVE $395
2 for $775

The best drive available today! Complete and ready to use, with free 2-drive cable! Free software for speeding up your access time from 35ms to 12ms! Optional software for increasing data storage from 55K to 67.8K on your first drive, and to 89K on your second drive! Full 120 day money back guarantee!

Cat No.	Description	Price
1375	Drive w/2 drive cable	$395.
1396	4 drive cable (alone)	49.
1938	Tracks 36-40 software	10.
1147	Verbatim diskettes, 10 for	$33

HAZELTINE TERMINALS

Standard interfacing to most systems!

Cat No.	Model	Description	Price
2167*	1400	53 key, 80 x 24 display, demb	$840
2168*	1500	Intelligent, 74 key, 80 x 24	$1195
2169*	1520	Intelligent, 81 key, 95 char.	$1588

ATARI

THE AFFORDABLE COMPUTER

The perfect computer for both home and business!

Sophisticated, yet simple. Designed by the expert in home computer based accessories. It won't become obsolete, because it has an expandable memory, advanced peripheral components, and comprehensive array of software modules. Whether you've had programming experience or not, you'll find yourself operating your Atari in no time at all! Features of the Model 400 include: 57 key keyboard, high-resolution color graphics, built-in TV modulator for direct connection with any standard color TV, 8K RAM, 8K BASIC ROM, plus more and more! The model 800 has all the features of the 400 plus two channel digital recorder, 8K RAM, (expandable to 48K) and 16K ROM, (expandable to 32K).

Cat No.	Description	Price
2172	Atari 400 Personal Computer	$550.00
2176	Digital Recorder for 400	90.00
2173	Atari 800 Personal Computer	$995.00
2174	Minidisk Drive for 800	750.00
2175	Line Printer for 800	600.00

8" Disks
$37 box of 10
3 boxes for $100

- *IBM compatible*
- *Single density*
- *Individually certified*

Cat No. 1145 Type 32-1000
Description
32 sector holes, 1-index hole

Cat No. 1146 Type 34-1000
interchangeable with IBM32, 3740, 3770, 3790, etc.

Verbatim

5¼ DISKETTES

The most popular diskette in the world!

$29/box of 10
2 for $55

Cat No.	Type	Description
1147	525-01	Soft, TRS-80
1148	525-10	10 hole hard, Apple, Northstar, H8
1149	525-16	16 hole hard, Micropolis

DISKETTE/DISK BINDERS

Organizes and protects your data files! 5¼" disks fit two per insert, 8" fit one per insert. Binders include 10 inserts.

Cat No.	Description	Price
1650	5¼" binder	$9.95
1651	8" binder	$9.95
1653	extra 8" inserts	.95
1652	extra 5¼" inserts	.95

FULLER ELECTRONICS

TRS-80 DUAL CASSETTE CONTROL CENTER $74.95

- *Aids and simplifies cassette loading!*
- *Allows you to listen to cassettes as they are loading!*

No more plugging and unplugging cassette cables! Saves wear and tear on your computer system, adds the flexibility that you need for level II data storage! Internal speaker allows you to listen to CSAVE and CLOAD, also allows you to listen to fast forward and rewind (CTR-41). Independent volume control lets you adjust monitor volume without affecting loading volume. Built-in LED's let you know status of one or both recorders. Now you can simultaneously CSAVE with both recorders; CLOAD with either recorder; cross tape with both recorders. Tested and reviewed in March '79 Creative Computing. Compact, only 4" x 3⅛" x 2⅛" high. Completely assembled and tested Wt 16 oz.

Cat No. 1973 For CTR-41
Cat No. 1974 For CTR-80 or CTR-21

FULLER ELECTRONICS RF-3

TRS-80 I/O & AUXILIARY CONTROL CENTER $39.95

- *Controls lights, video recorders, appliances thru your TRS-80!*
- *With complete application software and documentation*
- *Built-in relay protects your TRS-80 relay!*
- *Controls from fractions of a second to months!*

The most versatile TRS-80 accessory available today! Offers features found only in units costing much more. Now you can control lights when you're not home, at random times! Turn video recording equipment on and off at preset times, even months in advance. Internal beeper can be used to signal the end of a long sort. It also lets you know with one beep, two beeps, etc, exactly what part of the program you're in! Also allows you to check your RS-232 output without turning on the printer. Built-in internal relay protects and bypasses control current from your sensitive TRS-80 relay. Now you can even operate 6 or more cassette recorders safely for CSAVE. NOTE: 110VAC CONTROLLER IS NOT INCLUDED. A simple schematic and parts list is supplied. Size: 2" x 4" x 1⅝".

Wt 12 oz.
Cat No. 1972

SARGON II
Hayden

The champ of champs! Surpasses Microchess, and even Sargon I! Offers complex games, 7 levels of play, activity indicator, a special "hint" mode, plus more! The best chess program ever!
Cat No. 2082 TRS-80 $39.95
Cat No. 2083 APPLE II $29.95

TRS-80 FORTH
Programma

A compact threaded language. Vocabulary based, user may tailor the system to the needs and structure of a specific application. User defines new words, which may in turn define still more complex applications.
Cat No. 2133 TRS-80 L2, 16K $34.95

BARRICADE

Similar to breakout. A real time game, with options of speed, balls, angle, etc.
TRS-80 L1/L2 16K.
Cat No. 1362 $14.95

BRIDGE CHALLENGER

You and dummy play against the computer in regular contract bridge. Either you or comp sets up.
TRS-80 L2, 16K
Cat No. 1195 $14.95

TIELINE
Programma

Turns your TRS-80 into a timeshare mainframe, smart terminal, and allows you to send/receive BASIC and data via phone. Requires modem, RS232, 32K, DOS 2.1 or later.
Cat No. 2137 $6.95

ANDROID NIM II

Super improved version of NIM. Constant excitement!
TRS-80, L2, 16K
Cat No. 1686 $14.95

TRS-80 PIE
Programma

2 dimensional, cursor based editor. Features blinking cursor, cursor movement at up/down/left/right, plus tabs; character insert/delete; string search forwards and backwards; plus more and more! The best editor available for the TRS-80!
Cat No. 2134 TRS-80, L2, 16K $19.95

SYSCOP

Duplicates SYSTEM tapes (for backups).
TRS-80 L2, 48K.
Cat No. 1481 $9.95

STAR TREK III

The most advanced version we've seen!
TRS-80 L2, 16K
Cat No. 1541 $14.95

ADVENTURE

Explore an almost endless maze of treasures and pitfalls. Challenging and fun!
TRS-80 L2, 16K.
Cat No. 1723 $14.95

HUNT
Programma

A new class of game where you invent your own characters, rules, and so on. All the best of Adventure, Dungeon, Quest, Dragons all rolled into one!
Cat No. 2145 TRS-80 L2, 16K $19.95

LEVEL III BASIC

Gives your TRS-80 the power of a full size system. Disk commands, advanced editing, etc.
TRS-80 L2, 16K
Cat No. 1332 $49

NEWDOS+

Better than TRSDOS! Allows DIR while in basic, and then returning to basic, DIRCHECK, DISASSEM, EDTASM, and SUPERZAP! A must for disk owners!
Cat No. 1549 TRS-80, L2 + disk $99

VERBATIM CERTIFIED CASSETTES
2 for $4.95

Certified specifically for personal computers such as the TRS-80, Apple, Pet, etc. Splice-free, leaderless, with folding recording tabs.
Cat No. 1545

SOUNDWARE

Add music and sound effects to your programs. Complete with software and hardware. Installs in seconds.
Cat No. 1898 TRS-80, L2 $29.95

TRS-80 LOWER CASE MODIFICATION KIT $19

Modifies your machine to display both upper and lower case. Installs in minutes, requires drill, soldering iron and screwdriver. With complete instructions.
Cat No. 1550

LEEDEX

12" MONITORS

Black and white, high resolution. Accepts composite input. No kit shipments.
Cat No. 1304 **$139**

16K MEMORY ADD-ON
$75 2 for $140

For TRS-80, Apple, Exidy

Installs in minutes, no special tools required. Complete with detailed instructions. Wt 4 oz.

Cat No.	Description
1156	For TRS-80 keyboard unit
1156A	TRS-80 without buffered cable
1156B	TRS-80 with buffered cable
1156C	For Apple
1156D	For Exidy Sorcerer

DIRECT CONNECT MODEM
Novation

- *Answer/Originate!*
- *300 Baud!*
- *Designed specifically for personal computers!*

Model 41038, the top of the line in reliability, economy, convenience! Usable on either dial-up or leased telephone lines, the 41038 is Bell 103A or 113B compatible. Operates at the highest standards of -9dBm for transmit level, and -50dBm sensitivity. Meets all FCC requirements for direct line hookup, all units supplies with standard new modular "T-Plug". Low profile design complements the appearance of all computer systems. May be mounted on either wall or desk. Front panel has LED indicating RS232 interface status. Compact power pak plugs directly into wall socket, reduces heat and shock hazard.
Weight: 1 lb
Cat No. 3214* **$395**

Novation "Cat"

Acoustic Modem

The first quality compact modem designed specifically for the small computer user. Designed to transmit data over standard telephone lines, the Cat has many uses for business and hobbyist. Allows one terminal or computer to interact with others, permitting one access to data from remote locations. Hobbyists can exchange data and programs. Compatible with any Bell 103 modem and, of course, all other Cats! Also perfectly compatible with the new Radio Shack modem (it's the same unit). Data exchange can occur at any speed up to 90 charsec. Low profile case features switches for mode selection and operation, also for LED's displaying unit status. Acoustic self-test is standard. Complete and ready-to-use. Requires 110 VAC, 60Hz.
Cat No. 1480 **$169**

HOW TO ORDER

Pay by check, Mastercharge, Visa, or COD. Charge orders please include expiration date. Foreign pay in U.S. funds. Order by phone or mail, or at our retail. MINIMUM ORDER $10. please include phone number and magazine/issue you are ordering from. Prices valid thru last day of cover date. SHIPPING: USA: add $2.00 for the first 2 lbs. For ground add 35¢ for addt'l lb. For air add 75¢ for addt'l lb. FOREIGN: surface: add $3.00 for first 2 lbs, 60¢ per addt'l lb. Air: add $11.00 for first 2 lbs, $5 for each addt'l lb. CODs $1 addt'l. Guaranteed satisfaction for 120 days or your money back! Not responsible for typographical errors. Some items subject to prior sale. We reserve the right to limit quantities.

SEND FOR FREE CATALOG FEATURING:
Computers and accessories, disk drives, printers, integrated circuits, LEDs, semiconductors, books, software, connectors, plus more and more! The widest selection at the lowest prices! Circle our reader service number or phone/write today for your copy!

19511 Business Center Dr. Dept. V3 Northridge, Ca. 91324

TRS-80 SERIAL I/O

• Can input into basic • Can use LLIST and LPRINT to output, or output continuously • RS-232 compatible • Can be used with or without the expansion bus • On board switch selectable baud rates of 110, 150, 300, 600, 1200, 2400, panty or no panty odd or even, 5 to 8 data bits, and 1 or 2 stop bits D.T.R. line • Requires +5, -12 VDC • Board only $19.95 Part No. 8010, with parts $59.95 Part No. 8010A, assembled $79.95 Part No. 8D10 C. No connectors provided, see below

EIA/RS-232 connector Part No. DB25P $6.00 with 9', 8 conductor cable $10.95 Part No. DB25P8

3' ribbon cable with attached connectors to fit TRS-80 and our serial board $19.95 Part No. 3CA840

GAME PADDLES & SOUND

Includes: 2 game paddles, interface, software, speaker, power supply, full documentation including: schematics, theory of operation, and user guide; plus 2 games on cassette (Pong and Starship War). $79.95 Complete Part No. 7922C

16K RAMS

For the Apple, TRS-80 or Pet $8 each Part No. 4116/2117.

DIGICOM DATA PRODUCTS INC. Series 312 Acoustic Coupler

300 BAUD Originate, Part No. AC3122, $219.95. 300 BAUD Answer, Part No. AC3122, $219.95. 300 BAUD Answer/Originate, Part No. AC3123, $229.95.

LIGHT-PEN For Your TRS-80

Your TRS-80 Light-Pen is a carefully engineered instrument and with the proper care will give satisfactory use and many years of service. Part No. TRS80LP $24.95

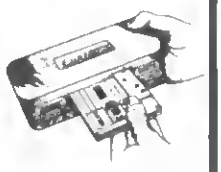
SYSTEM EXPANSION from LNW Research

• Serial RS232C/20 mA I/O • Floppy controller • 32K bytes memory • Parallel printer port • Dual cassette port • Real-time clock • Screen printer bus • Onboard power supply • Software compatible • Solder mask, silk screen. PC board and user manual, Part No. LNW80, $69.95.

DISKETTES

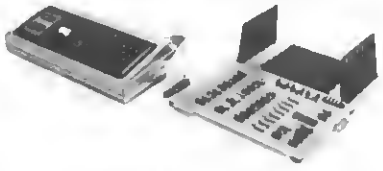
Box of 10, 5" $29.95, 8" $39.95. Plastic box, holds 10 diskettes, 5" - $4.50, 8" - $6.50.

CASSETTE TAPE ERASER



Removes recordings in one second! ERASER-8 $19.95

S-100 INTERFACE



AN S-100 bus Adapter—Motherboard for the TRS-80. Kit, Part No. HUH81DLXK, $295.95. Assembled, Part No. HUH81DLXA, $375.95

NOW! A FULL SUPPORT SYSTEM FOR TRS-80

• 32K of RAM • EPROM firmware • Disk control • Data acquisition • Parallel I/O • Serial I/O • Plug into GPA's Motherboard. GPA's quality design includes • 6-44 pin edge connectors • +5V, -5V, +12V, -12V external power supply required • Active termination. The Motherboard, Part No. GPA80, is only $149.95.

TAKE ADVANTAGE OF GPA-EXPANSION CARDS FOR THE GPA80

Memory cards: Now with Fortran compilers available for your TRS-80, additional expansion memory is a must! Card with sockets only, Part No. GPA801, $119.95. Card with 16K of 4116 Dynamic Ram, Part No. GPA802, $224.95. Card with 32K of 4116 Dynamic Ram, Part No. GPA803, $329.95. All cards come equipped with sockets to accomodate 32K of Ram.
EPROM firmware card. Put those valuable subroutines in firmware. Don't waste time loading and unloading tapes and disks. For 2708 or 2716 EPROMS, Part No. GPA806, $79.95.
Serial I/O card. Here's what you've been asking for, a full serial terminal interface, with RS-232C or 20 mA. Current loop. Input/output capabilities. Part No. GPA807, $79.95.
Parallel I/O Card. Control functions in the outside world, monitor and store real time events. Two parallel output ports. Dip switches select ports (0-254). Part No. GPA808, $79.95.

LEEDEX MONITOR

12" Black and White • 12 MHz Bandwidth • Handsome Plastic Case • $139.00

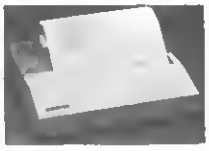
FLOPPY DISK STORAGE BINDER

Three-ring binder comes with ten transparent plastic sleeves which accommodate either twenty, five-inch or ten, eight-inch floppy disks. Binder & 10 holders $14.95 Part No. 8800; Extra holders 95¢ each. Part No. 800.

DISK JACKET™

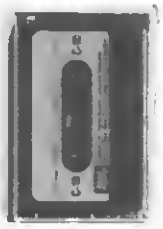
Holds two 5-1/4 inch diskettes and will fit any standard three ring binder $9.95/10 Pack.

TRENDCOM PRINTER



• 40 characters per second • 4-7/16 inch wide thermal paper • Graphics (TRENDCOM 100): 480 sevendot print positions per line. TRENDCOM 100, Part No. TRC0100, $495.95. TRENDCOM 200, Part No. TRC0200, $375.95. **Interface** for TRS-80, Part No. T80A $45.95. For Apple II, Part No. TRCAII, $75.95. For PET, NO. TRCP2, $79.95. For Scoccerer, TRCSR1 $45.95.

DIGITAL CASSETTE



5 min. each side. Box of 10 $9.95

SARGON: A Computer Chess Program

Features the complete program that won the 1978 West Coast Computer Faire Tournament. Part No. 00603 — TRS-80 Level II. Part No. 00604 — Apple II (24K). $19.95

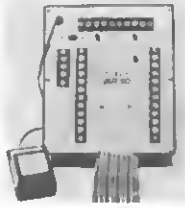
SOUND EFFECTS AND MUSIC FOR YOUR COMPUTER

SOUNDWARE is a complete system. It includes a speaker/amplifier unit with volume control, earphone jack, and connectors. It boasts excellent tone quality yet is small and convenient to use. Add batteries, plug it in, and play. One year warranty. SOUNDWARE package (includes INTRO to SOUNDWARE programs) PET (8K), Part No. 20003, $29.95. TRS-80 Level II (16K), Part No. 20002, $29.95. Compucolor II (8K), Part No. 20001, $39.95. INTRO to SOUNDWARE programs only PET and TRS-80, Part No. 20005, $14.95. Compucolor II Part No. 20006, $19.95.

COMPUCRUISE

$129.95; with cruise control $169.95

THE TELESIS VAR-80 INTERFACE UNIT



For the TRS-80 with Level II Basic • Provides 8 outputs • Provides 8 inputs • 2 ft. of interconnecting cable w/connector • Plugs directly into TRS-80 • Power supply provided • Assembled and tested. Part No. VAR80, Introductory price $109.95.

To Order: Mention part no. description, and price. In USA shipping paid by us for orders accompanied by check or money order. We accept C.O.D. orders in the U.S. only, or a VISA or Master Charge no., expiration data, signature, phone no., shipping charges will be added. CA residents add 6.5% for tax. Outside USA add 10% for air mail postage and handling. Payment must be in U. S. dollars. Dealer inquiries invited. 24 hour order line (408) 448-0800 ↙26

Send for FREE Catalog . . . a big self-addressed envelope with 41¢ postage gets it fastest!

ELECTRONIC SYSTEMS Dept. 80 P. O. Box 21638, San Jose, CA USA 95151

If you enjoy driving, you're going to get a COMPUCRUISE. Once you see what it can do, you just won't be able to live without it.

This gadget fits into most dashbdards . . . no strain even in a tiny sports car like the Mazda RX-7 . . . and once you have it, every trip is like flying a 747. The darned thing tells you the time, how fast you're going, how far you've been on this trip or since the last regassing, how many miles per gallon you're getting, either at the instant or the average on the trip . . . or gallons per hour at the moment or for the trip . . . temperature outside . . . inside (or coolant temperature, if you prefer) . . . oh, it has an elapsed time for the trip, a stop watch, lap time, an alarm . . . how much further for your trip, how many gallons more the trip will take, how much longer for the trip at your present average speed . . . yes, it gives you your average speed for the trip. You prefer it in metric, no strain . . . litera remaining, etc. Did we mention that it also has cruise control either at a speed set on the control board or at whatever speed you are traveling? The Compucruise will keep you busy and entertained during any trip . . . telling you more than you will ever want to know.

The Compucruise is not difficult to install . . . though it does connect to everything except the cigarette lighter. Until you've tried computerized travel, you haven't found out how much fun driving can be. It will work on any car not having fuel injection . . . and there is a front-wheel drive accessory gadget available for only $4.40-#P001 (regularly $5.50).

The price for the Compucruise is regularly $199.95 . . . and a bargain at that price. We'll sell you one of these fantastic gadgets for $159.95 with cruise control (Model 44-#P002), and $127.95 without (Model 41-#P003). Send money . . . and start having fun!

AUM'S

MAIL ORDER MICROS

Dept MBO • PO Box 427 • Marlboro NH 03455

Phone: (603) 924-3041

Steel Stopper

Ever had your car stolen?
The first reaction is one of disbelief . . .
. . . you *know* it was *right there*!

What you want is a modern combination lock on your ignition . . . The Steal Stopper. It's easy to install and almost impossible to defeat. You can by-pass it, if you want, for parking attendants or a car wash. Other than that, you set up a secret four digit code and only will then be able to start the car . . . even if you leave the keys in the ignition.

This protection retails for $50 . . . but we have a special for you at $39.95. Don't procrastinate. Order #P004.

Note: This product works best on Detroit cars. Mazda RX7 owners must order additional module, #P008, which costs $8. The Steal Stopper can be modified for Mercedes, Porsche, Ferrari, or other high performance European cars by returning unit to manufacturer with $3. They promise quick modification and return.

PROCESSOR TECH HARDWARE

9 **Processor Tech Video Display Module**—Memory mapped video for S-100, excellent condition. #D009-$144 each.

PROCESSOR TECH SOFTWARE

2 **Processor Tech Extended Disk BASIC**—This is full disk BASIC on 8" disk for HELIOS II disk controllers with PTDOS and greater than 16K. #D015-$80 each.

1 **Extended Disk BASIC on cassette**—This is the same as previously mentioned for the Disk BASIC from Processor Tech. Needs more than 16K. #D017-$72.

8 **Extended Cassette BASIC**—This includes all file operations, advanced functions for doing more than playing games; for SOLOS, CUTTER, and CONSOL Monitors. #D016-$22 each.

19 **BASIC 5 from Processor Tech**—This is a simple BASIC for a SOLOS, CUTTER, or CONSOL Monitor and 8K of RAM. #D013-$11.60 each.

4 **Processor Tech GAMEPAC** for above BASIC—Various simple games. #D014-$11.60 each

ASTROLOGICAL COMPUTER AND 4-FUNCTION CALCULATOR

ASTRO®

Gives you a fascinating look into your personality traits. Compares 16 different combinations. Matches your astrological influences to any day—past, present, or future. Gives in-depth analysis of your compatibility with your boss, your spouse, your lover, your child. Regularly $49.95. #P0020—$39.95 SPECIAL PRICE.

PROCESSOR TECH SOL

4 **SOL Computers**—8K RAM Monitor, S-100, excellent condition. #D004-$980 each.

19 **TREK-80** on cassette for SOL—This is one of the best real time space games available today; needs 8K. #D005-$11 each.

9 **Electric Pencil** on cassette for SOL—Word processor, needs 8K. #D006-$80 each.

MICRO TERM ACT TERMINAL

2 **MicroTerm ACT Terminal**—Need a video monitor, up to 600 Baud, good condition. #S035-$200 each. SPECIAL PRICE.

COMPUCOLOR HARDWARE

2 **Compucolor Computer 8001** (use as computer or 75 MHZ Color Monitor)—8K RAM, BASIC and DOS in ROM, good condition. #S025-$1500 each. SPECIAL PRICE.

2 **Compucolor MiniFloppy**—5¼ inch, good condition. #S026-$525 each. SPECIAL PRICE.

4 **Compucolor 8K RAM card**—Static RAM, good condition. #S027-$200 each. SPECIAL PRICE.

3 **Compucolor Floppy Tape Drive**—Uses eight track cartridges, good condition. #S028-$70 each. SPECIAL PRICE.

PANASONIC TAPE DECKS

Panasonic RS261 US Stereo Cassette Decks—with auto-stop, record level adjust, VU meters, used condition; all have had heads replaced and aligned #T001-$50.

Panasonic RS260 US Stereo Cassette Decks—same as above but also has bias switch for chrome tapes. #T002-$50.

POLYMORPHIC HARDWARE

6 **PolyMorphic Video Terminal Interface**—Memory mapped video for S-100 bus, good to excellent condition. #S044-$150 each. SPECIAL PRICE

1 **PolyMorphic Cassette Interface**—Plugs into Poly CPU only, new. #D047-$29.60 each.

1 **PolyMorphic S-100 Cabinets**—Nice 5-slot S-100 mainframe, good to excellent condition. #D049-$248 each.

POLY-88 ACCESSORIES

3 **Parallel Keyboard**—Good condition. #S001-$50 each. SPECIAL PRICE.

1 **Software on 5¼"**—This is system software that requires a PolyMorphic Disk Controller in a System 88 Cabinet. #D002-$100.

1 **Electric Pencil**—#D003-$80.

ALS-80

14 **ALS-80 Operating system**—This system requires 12K RAM from D000 to FFFF; as well as either the SOLOS or CUTTER monitor; it includes an Assembler/Editor. #D018-$11.60 each.

NORTH STAR HARDWARE

1 **North Star Floating Point BASIC card**—With special BASIC, new. #D0059-$287.

2 **North Star Floating Point BASIC card (kit)**—With special BASIC, S-100, new. #D0060-$207.

8 **North Star Floppy Disk Controller card**—Single density, S-100, new. #D0061-$248.

120 **S-100 Edge Connector**—Gold Contacts, new. #D050-$2 each.

6 **Extender Card for S-100 (kit)**—New. #D051-$24 each.

BALLY GAMES

2 Bally VideoCode Cassettes—They consist of two games: Speed Math and Bingo Math. #D029-$16 each

ABACUS

112 Abacus Paperweight—Hefty, brass, excellent condition. #S024-$3 SPECIAL PRICE.

MUSIC

2 **Software Technology S-100 Music system** on cassette—This is an S-100 Music system: contains the proper hardware. #D0058-$19.60 each.

COMPUTER TRAINER

1 **IASIS COMPUTER-IN-A-BOOK**—8080 Microcomputer, comes built into training manual, excellent condition. #D020-$240

ICOM DISK DRIVE ACCESSORIES

1 **ICOM Dual Disk Drive**—Single density, 512K storage, S-100 controller, includes CP/M ROM, good condition. #S030-$1500. SPECIAL PRICE.

1 **ICOM PROM and 8" Disk for SOL FDOS**—This disk requires an ICOM S-100 Disk Controller installed in an S-100. #D031-$160.

1 **ICOM CP/M on 8" Disk for S-100**—Requires an ICOM S-100 controller in an S-100 cabinet, #D032-$100.

2 **ICOM FDOS-II on 8" Disk for S-100**—Requires an ICOM S-100 controller in an S-100 cabinet, no documentation. #D033-$180 each.

1 **ICOM FDOS-II on 5¼" Disk for S-100**—Requires an ICOM S-100 Mini-Floppy Controller in an S-100 cabinet. #D034-$168.

MOM'S
MAIL ORDER MICROS

20% OFF
Inventory Clearance SALE

IMSAI HARDWARE

1 **IMSAI 8800 Mainframe S-100**—Excellent condition. #D0087-$839.

1 **IMSAI 80/15 S-100 Development System**—Partially assembled, needs a CPU card, excellent condition. #S0088-$525 as is. **SPECIAL PRICE.**

1 **IMSAI 80/15 S-100 Development System**—Kit, mainframe cover missing, needs a CPU card, excellent condition. #S0089-$500 as is. **SPECIAL PRICE.**

2 **IMSAI 4K RAM card**—S-100, good condition. #D0055-$89.60

IMSAI SERIAL I/O CARDS

1 **IM[illegible] Serial I/O card 2** SOLD OUT [illegible] ports, full RS-232 [illegible] J, good condition. #D0[illegible]188.

5 **IMSAI Serial I/O card 2-2 (kit)**—Two serial ports, full RS-232 control, S-100, new. #D0091-$124.

2 **IMSAI Serial I/O card 2-1 (kit)**—One serial port, full control RS-232 control, S-100, new. #D0092-$100.

IMSAI PARALLEL I/O CARDS

1 **IMSAI Parallel I/O card 4-4**—Four parallel ports, S-100, excellent condition. #D0093-$186.

1 **IMSAI Parallel I/O card 4-1 (kit)**—One parallel port, S-100, new. #D0094-$74.40.

IMSAI SOFTWARE

2 **IMSAI IMDOS V2.02** on 8" Disk for S-100—No documentation, but this is apparently IMSAI's version of CP/M for S-100 systems with an IMSAI Disk Controller. #D0056-$96 each.

2 **IMSAI BASIC 9A**—This BASIC is for an S-100 system with a TARBELL cassette interface. #D0057-$22

1 **TARBELL Cassette Interface (kit)**—Kansas City Interface, Tarbell Phase encoding, S-100, new. #D0064-$96

PHONE INTERFACE

6 **Novation Modem #3102A**—Connects to any phone, originate only, good condition. #S021-$165 each. **SPECIAL PRICE.**

2 **Novation Modem #31** SOLD OUT [illegible] to any ph[illegible] ite, good condition. #[illegible] 2-$200 each. **SPECIAL PRICE.**

1 **Novation Modem #43**—Connects to any phone, originate only, good condition. #S023-$165. **SPECIAL PRICE.**

VECTOR GRAPHIC S-100

3 **Vector Graphic ROM/RAM card**—12K empty ROM sockets, 1K RAM, excellent condition. #D0078-$119.60.

5 **Vector Graphic Analog Interface**—Allows hobbyist to interface analog experiments, S-100, new. #D0079-$79.20.

1 **V[illegible]raphic Analog In[illegible]** SOLD OUT [illegible]e, S-100, new. #[illegible] 55.60

TDL ZAPPLE

2 **TDL Z-80 8K BASIC**—This is for a Z-80 system with a ZAPPLE Monitor. #D0071-$30.

1 **TDL [illegible]2K Moni[illegible]** SOLD OUT [illegible]tart a system with. #[illegible] $16.

3 **TDL ZAP 1K Monitor**—Simple monitor. #D0073-$12.

NEXT MONTH . . . WATCH THIS SPACE FOR 20% (OR MORE) DISCOUNTS ON POPULAR BRAND NAME SOFTWARE FROM MOM'S.

BOOK CLEARANCE UP TO 50% OFF

Take a Chance with Your Calculator (Lithium-publisher) #BK1002—was $8.95, now $4.50.

Chemistry with a Computer (Educomp-publisher) #BK1010—was $9.95, now $5.00.

Computer Dictionary (Camelot-publisher) #BK1018—was $5.95, now $3.00.

FORTRAN Programming (Camelot-publisher) #BK1019—was $7.95, now $4.00.

FORTRAN Workbook (Camelot-publisher) #BK1020—was $4.95, now $2.50.

A Quick Look at BASIC (Camelot-publisher) #BK1043—was $4.95, now $2.50.

TDL SOFTWARE-DISK

1 **TDL FDOS & SuperBASIC on 8" Disk**—This requires an ICOM Disk Controller and at least 20K of memory, plus a ZAPPLE Monitor in an S-100 Cabinet (Altair, IMSAI, etc.). #D0065-$137.

1 **TDL System Software on 5¼" disk**—This set of system software requires a North Star Disk Controller, a TDL Systems Monitor Board I, and consists of 12K BASIC, Relocator/Linking Loader, Z-80 Editor, and Text Processor. #D0066-$183.

1 **TDL System Software on 5¼" disk**—This is the same as above, but does not require the Systems Monitor Board I. #D0067-$200.

1 **TDL System Software on 5¼" disk**—Again, as above, but requires a HELIOS Disk Controller and the TDL Systems Monitor Board II (not I). #D0068-$183.

HONEYWELL

15 **Honeywell ASR-33 Communications Consoles** with TTY, paper tape reader and punch. Used, working when removed from service. Shipped freight collect or you pick up. Weight 300 lbs. $395. Order #P006.

MOUNTAIN HARDWARE

7 **Mountain Hardware AC Controller**—Remote AC outlet control, S-100, new. #D040-$100 each.

4 **Mountain Hardware remote outlet**—Remote module for above, two channels, new. #D041-$72 each.

HEURISTICS SPEECH LAB

1 **Heuristics Speech Lab**—S-100, used, fair condition. #S042-$100 as is. **SPECIAL PRICE.**

4 **Heuristics Speech Labs**—S-100, new. #D043-$151 each

SHUGART MINIDISK DRIVES

1 **Shu[illegible]rt MiniDisk Drives**—[illegible] SOLD OUT [illegible] and power supply, [illegible]ble for North Star, g[illegible]ondition #S036-$320. **SPECIAL PRICE.**

4 **Shugart MiniDisk Drives**—No cabinet, good condition, #S037-$300 each. **SPECIAL PRICE.**

4 **Power Supplies** for above—good condition #S038-$30 each. **SPECIAL PRICE.**

Prices include 20% discount. **SPECIAL PRICE** includes more than 20% discount.

Quantities are limited, immediate refund if ordered item is no longer available. *Phone answered by machine. Orders taken with credit cards. Questions answered by mail.

TERMS: FOB Marlboro, NH USA. Limited stock; everything guaranteed as described; you pay postage on returns. PRINT orders clearly. Minimum order $10 plus $2.50 shipping and handling charge in USA only. DOUBLE THAT ELSEWHERE. Orders over $50 add 5% for shipping in USA, 10% elsewhere (we will refund excess). Orders shipped UPS or insured mail only. No CODs please. Send US funds by check or money order. For credit card purchases, add 4%, list AE, MC or VISA, number, and expiration date. Mail to MOM's, Department M80, PO Box 427, Marlboro NH 03455.

Condition of Inventory:
New = original container
Excellent = new, but not in original container
Good = tested or used in store

Qty	Catalog #	Description	Unit Price	Total

Delivery: 3 to 6 weeks.

Enclosed $ _____

Bill: AE MC VISA

Card no. _____ Exp Date _____

Name _____

Address _____

City _____ State _____ Zip _____

Ship UPS ☐ Insured mail Signature _____

MOM'S
MAIL ORDER MICROS
Dept M80 • PO Box 427 • Marlboro NH 03455
Phone: (603) 924-3041

Shipping & Handling _____

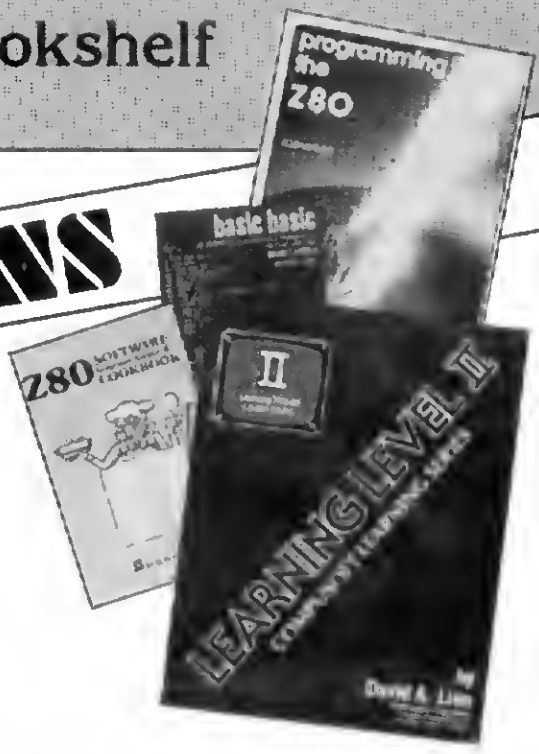
Credit Card (+4%) _____

Total _____

✓ Reader Service—see page 147

80 microcomputing™ bookshelf

NEW ADDITIONS



● **PROGRAMMING THE Z-80** – BK1122 – by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK** – BK1045 – by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $14.95.*

● **LEARNING LEVEL II** – BK1175 – by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **BASIC BASIC (2ND EDITION)** – BK1026 – by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $9.45.*

● ***MICROCOMPUTING* CODING SHEETS** *Microcomputing*'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

● **HOBBY COMPUTERS ARE HERE!** – BK7322 – If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, and the basics of programming. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

● **THE NEW HOBBY COMPUTERS** – BK7340 – This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards . . . and much, much more! Don't miss this tremendous value! Only $4.95.*

● **HOW TO BUILD A MICROCOMPUTER – AND REALLY UNDERSTAND IT** – BK7325 – by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*

● **TOOLS & TECHNIQUES FOR ELECTRONICS** – BK7348 – is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.*

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 microcomputing bookshelf

●THE BASIC HANDBOOK – BK1174 – by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

●ADVANCED BASIC – BK1000 – Applications and problems by James Coan is for those who want to extend their expertise with BASIC. Offers advanced techniques and applications. $7.95.*

●PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM – BK1009 – Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $9.95.*

●PAYROLL WITH COST ACCOUNTING – IN BASIC – BK1001 – by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941. $20.00.*

●LOW-COST, PERSONAL COMPUTER-BASED INVESTMENT DECISION SYSTEMS – BK1101 – Use this guidebook by Man-Computer Systems, Inc.'s president, Jerry Felsen, to develop inexpensive personal computer systems that can help you make better investment decisions. $15.00.*

●HOW TO MAKE MONEY WITH COMPUTERS – BK1003 – In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

●SOME COMMON BASIC PROGRAMS – BK1053 – published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $12.50 paperback.*

●WHAT TO DO AFTER YOU HIT RETURN – BK1071 – PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

●BASIC COMPUTER GAMES – BK1074 – Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

●SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition) – BK1073 – by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to 80 Microcomputing Bookshelf ● Peterborough NH 03458. Be sure to include check or detailed credit card information. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

80 PREVIEW

WHAT TO LOOK FOR IN THE APRIL 80

PENCIL v SCRIPSIT

The reigning champion of word processing software, Michael Shrayer's Electric Pencil, has a challenger—Scripsit from Radio Shack. Next month we have an in-depth evaluation of both packages. Find out what they can and can't do.

REVERSE OEDIV

More than an hour or two at the keyboard of your system is liable to result in tired fingers and bleery eyes. We don't know what to do about your fingers, but Reverse Video can help ease the strain on your eyes. Black lettering on a white screen makes working with a computer like reading a book.

BUILD A LIGHT PEN

Your keyboard is not the only way of interacting with the computer. With next month's 80 you can build a simple Light Pen, and just point at the screen! Complete software is included for using the device, both in BASIC and Assembly Language.

ASSEMBLY LANGUAGE

If you know what 50H means then this new column is not for you. For the remaining 80 percent of our readers we are starting an Assembly Language column in the April edition. Through worked example and clear explanation this series will cover Assembly Language A-Z. By the way, 50H is '80' in hexadecimal.

MORK & MINDY MONITOR

Tired of carrying your Radio Shack monitor from place to place? With this article you can find out how to use any television set as a computer monitor.

TRUE CONFESSIONS

Purchase of a TRS-80 can produce a dramatic change in both life-style and character. Staying up into the wee small hours of the morning, talking in BASIC to strangers in the street . . . Follow the sad tale of a member of Computers Anonymous. It could happen to you!

ALL THIS PLUS MUCH MORE!

80 ADVERTISERS

Ask for Instant Software at a computer store near you.

Alabama

Anderson Computers
3156 University Dr., Huntsville

Computerland of Huntsville
3020 University Dr., Huntsville

Olensky Bros.
3763 Airport Blvd., Mobile

Arizona

Hem Sheck
450 8-A N. 16th St., Phoenix

Millets TV & Radio
821 East Broadway, Mesa

California

Byte Shop of Fairfield
87 Marina Center St., Suisun City

Byte Shop
8038 Clairmont Mesa Blvd.,
San Diego

Byte Shop of Mt. View
1415 West El Camino Real, Mt. View

Byte Shop of Sacramento
6041 GreenBack Ln., Citrus Heights

Capital Computer Systems
3396 El Camino Ave., Sacramento

Computers Made Easy
819 East Ave. Q-9, Palmdale

Computer Store of San Leandro
701 MacArthur Blvd., San Leandro

Computer World
8791 Westminster Ave., Westminster

Computerland
16720 S. Hawthorne, Lawndale

Computerland of San Francisco
117 Fremont St., San Francisco

Computerland of W. LA
6840 La Cienega Blvd., Inglewood

Coast Electronics
3118 No. Main St., Morro Bay

Hobby World
19511 Business Ctr. Dr., Unit 6
Borthridge

I.C.E. House Inc.
398 North E. St., San Bernardino

Microsun Computer Center
2989 North Main St., Walnut Creek

Opamp/Technical Books
1033 N. Sycamore Ave., Los Angeles

O.I. Computers, Inc.
15618 Hawthorne Blvd., Lawndale

Radio Shack Dealer
8250 Mire Mesa Blvd., San Diego

Santa Rosa Computer Center
604 7th St., Santa Rosa

Silver Spur Elect. Comm
13552 Central Ave., Chino

The Computer Store
820 Broadway, Santa Monica

Colorado

Byte Shop
3464 S. Acoma St., Englewood

Colorado Computer Systems
311 W. 74th Ave., Westminster

Computerland of North Denver
8749 Wadsworth Blvd., Arvada

The Computer Broker
2300 Welton St., Denver

The Computer Store
2300 Welton St., Denver

Connecticut

American Business Computers
454 Thames St., Groton

Computerlab
130 Jefferson, New London

Computerland
1700 Post Rd., Fairfield

Computer Works
1439 Post Rd. E. Liberty Plaza,
Westport

D.C.

The Program Store
4200 Wisconsin Ave., N.W.,
Washington, D.C.

Florida

Adventure International
200 Bald Cypress Ct., Longwood

AMF Electronics
11146 N. 30th St., Tampa

Boyd-Ebert Corporation
1328 West 15th St., Panama City

Computer Center
6578 Central Ave., St. Petersburg

Computerland of Ft. Lauderdale
3963 N. Federal Hwy., Ft. Lauderdale

Computerland of Jacksonville
2777-6 University Blvd. W.
Jacksonville

Computer Shack
3336 Beach Blvd., Jacksonville

Curtis Waters Enterprises
236 Talbot Ave., Melbourne

Heath Kit Electronic
4705 W. 16th Ave. Center, Hialeah

Sound Ideas
2201-C N.W. 13th, Gainesville

Ukatan Computer Store
Airport Rd., Destin

Williams Radio & TV Inc.
2062 Liberty St., Jacksonville

Georgia

Atlanta Computer Mart
Atlanta

Computerland of Atlanta
2423 Cobb Parkway, Smyrna

Hawaii

Computerland of Hawaii
567 N. Federal Hwy., Honolulu

Radio Shack Assoc. Store
1712 S. King St., Honolulu

Idaho

Electronic Specialists
8411 Fairview Ave., Boise

Illinois

Bloomington Normal
Computer Works
124 E. Beaufort, Normal

Computerland
4807 North Sterling, Peoria

Computer Station
3659 Nameoki Rd., Granite City

Midwest Micro Computers, Inc
708 S. Main St., Lombard

Indiana

Computer Center of South Bend
51591 US 31 North, South Bend

Iowa

Memory Bank
4128 Brady St., Davenport

Kansas

Central Kansas Computers
8 S. Broadway, Herington

Louisiana

Computer Shoppe Inc.
3225 Danny Park, Suite 222, Metairie

Maryland

Jack Fives Electronics
4608 Debilan Circle, Pikesville

The Comm Center
9624 Ft. Meade Rd., Laurel

Massachusetts

ComputerCity
5 Dexter Row, Charlestown

Computerland of Boston
214 Worcester Rd., Wellesley

Computer Packages Unlimited
244 W. Boylston St., West Boylston

Lighthouse Computer Software
14 Fall River Ave., Rehobeth

New England Electronics Co
879 Highland Ave., Needham

The Computer Store
120 Cambridge St., Burlington

Tufts Radio & Electronics
206 Mystic Ave., Medford

New York

Arleto Craft
314 Fifth Ave., NYC

Bits & Bytes
2800 Straight Rd., Fredonia

Computer Corner
200 Hamilton Ave., White Plains

Computer Factory
485 Lexington Ave., NYC

Computer House, Inc.
721 Atlantic Ave., Rochester

Computerland of Nassau
79 Westbury Ave., Carle Place

Computer World
519 Boston Post Rd., Port Chester

Comtek Electronics, Inc.
2666 Coney Island Ave., Brooklyn

Comtek Electronics, Inc.
Staten Island Mall
Store 220A, Staten Island

Home Computer Center
671 Monroe Ave., Rochester

Key Electronics
Schanectady

Mr. Computer
Imp. Plaza, Rte. 9, Wappingers Falls

Softron Systems
308 Columbia Turnpike, Rensselaer

The Computer Tree Inc.
409 Hooper Rd., Endwell

Upstate Computer Shop
629 French Rd., Campus Plaza
New Hartford

North Carolina

Byte Shop of Raleigh
1213 Hillsborough St., Raleigh

Ohio

Altair Business Systems, Inc.
5252 North Dixie Dr., Dayton

Astro Video Electronics
504 E. Main St., Lancaster

Cincinnati Computer Store
4818 Interstate Dr., Cincinnati

Computerland
4579 Great Northern Blvd.,
N. Olmstead

Computerland
1288 Som Rd., Mayfield Heights

Computer Store of Toledo
18 Hillwyck Dr., Toledo

Forbees Microsystems Inc.
35 N. Broad, Fairborn

Microcomputer Center
7900 Paragon Rd., Dayton

Micro-Mini Computer World
74 Robinwood, Columbus

21st Century Shop
16 Convention Way, Cincinnati

Oklahoma

Vern Street Products
Radio Shack Dealer
114 W. Taft St., Sapulpa

Oregon

Computerland of Portland
12020 S.W. Main St., Tigard

Computer Pathways Unlimited, Inc
2151 Davcor St. S.E., Salem

Pennsylvania

Artco Elect.
302 Wyoming Ave., Kingston

Artco Elect
Back Mountain Shop. Ctr.
Shavertown

Computer Workshoppe
3848 William Penn Hwy, Monroeville

Computerland of Harrisburg
4644 Carlisle Pike, Mechanicsburg

Erie Computer Co.
2127 West 8th St., Erie

Personal Computer Corp
24-26 West Lancaster Ave., Paoli

Personal Computer Corp
Frazer Mall, Lancaster Ave., Frazer

Michigan

Computer Center
28251 Ford Rd., Garden City

Computer Connections
38437 Grand River, Farmington Hills

Computerland of Grand Rapids
2927 28th St. S.E., Kentwood

Computerland of Rochester
301 S. Livernois, Rochester

Computerland of Southfield
29673 Northwestern Hwy., Southfield

Computer Mart
560 W. 14 Mile Rd., Clawson

Hobby House
1035 W. Territorial Rd., Battle Creek

Ye Olde Teacher Shoppe
1823 Willmyre St., Ypsilanti

Minnesota

Computerland of Hopkins
11319 Hwy F., Hopkins

Zim Computers
5717 Xerxes Ave. N. Brooklin Center

Mississippi

Dyer's, Inc
200 E. Main St., West Point

Missouri

Computervan, Inc.
51 Florissant Oaks Shopping Center
Florissant

Consolidated Software
16501 Greenwald Court, Balton

Montana

Intermountain Computer
529 So. 9th St., Livingston

Personet Computer
121 Red Oak Dr., Carl Junction

The Computer Store
1216 16th St. W. #35, Billings

Nebraska

Computerland of Omaha
11031 Elm St., Omaha

Midwest Computer Co. Inc.
8625 I St., Omaha

Midwest Computer Co. Inc
4442 S. 84th St., Omaha

Midwest Computer Co. Inc
4403 S. 87th St., Omaha

Navada

Century 23
4566 Spring Mountain Rd., Las Vegas

New Hampshire

Bitsnbytes Computer Center
568 Pleasant St., Concord

ComputerCity
1525 S. Willow, Manchester

Portsmouth Computer Center
31 Raynes Ave., Portsmouth

New Jarssy

Computer Encounter
2 Nassau St., Princeton

Computerland
35 Plaza Rte #4, W. Paramus

Computer Mart of NJ
501 Rte. 27, Iselin

Radio Shack/J&J Electronic
Mansfield Shopping Ctr
Rt. 57 Allen Rd., Hackettstown

The Bargain Brothers
Glen Rec Shopping Center
216 Scotch Road, Trenton

New Mexico

Legey and Associates
2908 Tahiti Ct. N.E., Albuquerque

South West Computer Center
121 Wyatt Drive, Suite 7, Las Cruces

South Dskota

CB Radio Sheck
21st and Broadway, Yankton

Tannassee

Computerlab
671 S. Menden Hall Rd., Memphis

H & H Electronics Inc.
509 N. Jackson St., Tullahoma

Texas

Computercraft Inc.
3211 Fondren, Houston

Computer Port
926 N. Collig, Arlington

Houston Microcomputer Tech
5313 Bissonet, Bell Aire

Interactive Computers
7620 Gashwood Rd., Houston

K.A. Elect.
9090 Stemmons Frwy., Dallas

Pan American Elect. Inc.
1117 Conway, Mission

Ram Micro Systems
8353 Camp Bowie Blvd., Ft. Worth

Virginia

Home Computer Center
2927 Virginia Beach Blvd.
Virginia Beach

Southside Radio Comm.
135 Pickwick Ave., Colonial Heights

Washington

American Mercantile Co. Inc.
2418 1st Ave. S., Seattle

Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way

Personal Computers
S 104 Freya, Spokane

Ye Old Computer Shop
1301 G. Washington, Richland

West Virginia

The Computer Corner Inc.
22 Beechurst Ave., Morgantown

The Computer Store
Municipal Parking Bldg., Charleston

Wisconsin

Byte Snop Of Milwaukee
6019 West Layton Ave., Greenfield

Wyoming

Computer Concepts
617 W. 16th St., Cheyenne

Puarto Rico

The Microcomputer Store
1568 Ave. Jesus T. Pinero
Caparra Terrace

Guam

The Fun Factory
851 Marine Dr., Tamuming

Canada

CANADIAN DISTRIBUTORS:
Micron Distributing
409 Queen St., W. Toronto, Ont
M5V 2A5

Computerland of Winnipeg
715 Portage Ave., Winnipeg, Man.

Compumart
411 Roosevelt Ave., Ottawa, Ontario

Computer Mart, Ltd.
1055 Yonge St., Suite 208
Toronto, Ontario

Gelectia Computers
103rd Ave., Edmonton, Alberta

Micrometic Systems Inc.
101 8130 Park Rd., Richmond, B.C.

Micro Shack of W. Canada
333 Park Street, Regina Sask

Orthon Holdings Ltd.
12411 Stony Plain Road
Edmonton, Alberta

Total Computer Systems
Ajax, Ontario

England

Temays & Farr Ltd
4 Morgan St., London

Franca

Sideg
45 Rue de la Chapelle, Paris

Sivea s a
20, Rue da Laningred, Paris

Italy

HOMIC s.r.l.
Piazza De Angeli 1, Milano

West Germany

Electronic Hobby Shop
Kaiserstr. 20, Bonn

MicroShop Bodensee
Merkstr. 3, 7776 Markdorf

Australia

Computerware
62 Paisley St., Footscray VIC

Deforest Software
36 Glen Tower Drive
Glen Waverly, VIC

Softronics Micro Systems
Lindfield

Sure-Load Software
P.O. Box 26, Weston, A.C.T.

South Africa

Eddie Talberg
P.O. Box 745, Johannesburg

Instant Software Inc.

Peterborough, N.H. 03458 603-924-7296